FLY WITH EASE

Software Engineering Project

Submitted in partial fulfilment, as a part of curriculum of

B.Sc. (H) Computer Science

from

Shyama Prasad Mukherji College for Women

University of Delhi

New Delhi

Submitted By:

Anjali, 18075570004

Sakshi Jain, 18075570037

Vaishnavi Chaudhary, 18075570045

Under the supervision of,

Dr. Baljeet Kaur

Assistant Professor

Dept. of Computer Science

Shyama Prasad Mukherji College for Women,

University of Delhi

New Delhi

ACKNOWLEDGEMENT

In performing our project, we had to take the help and guidance of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much pleasure. We would like to show our gratitude to Dr. Baljeet Kaur for giving us a good guidance for this project throughout the numerous consultations. We would like to expand our deepest gratitude to all those who directly and indirectly guided us in writing this project.

This project consumed huge amount of work, research and dedication. Still, implementation would not have been possible if we did not have a support of many individuals. Therefore we would like to extend our sincere gratitude to all of them.

Many people, especially our classmates and team members itself have made valuable comments and suggestions on this proposal which gave us an inspiration to improve our work. We are immensely grateful to all who are involved in this project as without their inspiration and valuable suggestions it would not have been possible to develop the project within the prescribed time.

Anjali, 18075570004
Sakshi Jain, 18075570037
Vaishnavi Chaudhary, 18075570045

CERTIFICATE

This is to certify that the content of this project entitled, "FLY WITH EASE" by Sakshi Jain, Vaishnavi Chaudhary and Anjali and is the bonafide work of them submitted to Shyama Prasad Mukherji College, New Delhi for consideration in partial fulfilment of the requirement of Delhi University as a part of the curriculum of B.Sc. (H) Computer Science.

The original research work was carried out by them under my supervision in the academic year 2019-2020. On the basis of declaration made by them I recommend this project for evaluation.

Certified By –

Dr. Baljeet Kaur
Assistant Professor,
Department of Computer Science
Shyama Prasad Mukherji College,
University of Delhi

TABLE OF CONTENT

PROBLEM STATEMENT

"Fly with Ease" is a software installed by airlines in their narrow body aircrafts with personal video screens at every backseat. The objective of this software is to provide a comfortable and luxurious travelling time with various facilities including an array of entertainment, victuals (meal, drinks, and beverages), weather information, inflight route tracking and many other facilities. It allows the interaction between flyers and flight crew members as well.

ACTORS

1. Flyers
2. Flight attendants (air hostesses and stewards)
3. Administrator

FUNCTIONALITIES

- **Register Module:**
  In this module, all the actors can create their new account.

- **Login/Logout:**
  Actors have to enter their username and password to login their account accordingly and they can also logout their account from the software.

- **Select Language:**
  This module will give the opportunity to user to select their native language among several languages.

Functionalities (Only For Flyers)

- **Entertainment Section:**
  This module is for flyers entertainment and it contains a huge collection of movies, songs, and games. User can perform following functionalities in this module:-
  - Watch movies
  - Listen music
  - Play games
  - Search: - User can search the movies, songs and music by their name.
  - Add to Favourite: - User can add their favourite movies and songs in their favourite list so that they do not have to search them again and again.

- **Victuals Section:**
  In this flyer can place their order from the menu items.
  - Place Order: - User can place their order by just adding the items to the cart.
  - Pay Bill: - User can pay for their bills by entering the details of their debit or credit card and for authentication they received unique 4-digit otp code through sms.

- **Weather Prediction Section:**
  The primary objective of this module is to show the prevailing weather conditions hourly or even minute-to-minute by using location services to the flyers.

- **View Details:**
  User can view all the details of flight like departure time, number of stops etc.

- **Acquaint Requirements:**
  Flyers can message the air hostesses for their other needs like towels, blankets etc.

- **Review:**
  Flyers can rate the software and submit their comments in this section.

Functionalities (Only For Administrator)

- **Add Movies, songs and games:**
  This section is available only for the admin so that they can upload the latest movies, songs and games in the software, so that flyers can view.

- **Delete Movies, songs and games:**
  This section is available only for the admin so that they can delete the old movies, songs and games from the software

- **View Record of Orders:**
  In this section administrator can view all placed orders.

- **View Record of Reviews:**
  In this section administrator can view all the ratings and reviews given by passengers.

Functionalities (Only For Flight Attendants)

- **View Order:**
  This section is available for the air hostesses and for stewards so that they can view placed order given by flyers.

- **View Messages:**
  In this section, flight attendants can view the messages given by passengers and can accept the order of their miscellaneous items.

SOFTWARE LIFE-CYCLE MODEL

INCREMENTAL MODEL

The incremental build model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

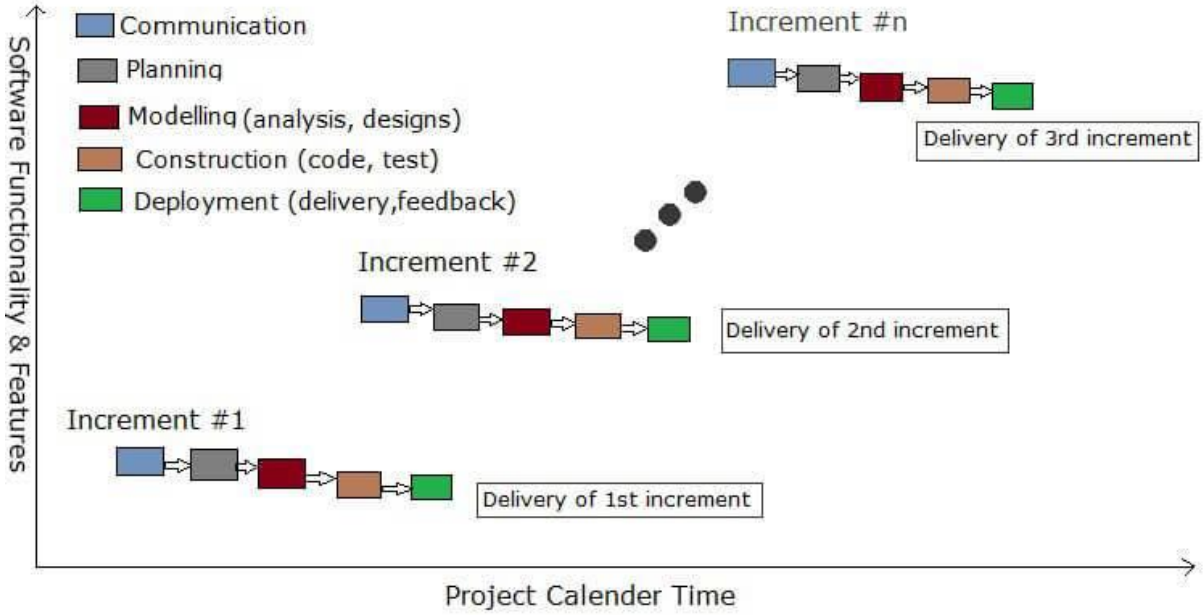


Fig.1: The Incremental Model

WHY INCREMENTAL MODEL IS USED FOR THIS PROJECT?

1. Generates working software quickly and early during the software life cycle.
2. More flexible – less costly to change scope and requirements.
3. Easier to test and debug during a smaller iteration.
4. Easier to manage risk because risky pieces are identified and handled during its iteration.
5. Each iteration is an easily managed milestone

PHASES IN THE INCREMENTAL MODEL

Communication:-
In this phase, the major task is to collect the requirements which leads us to find the exact need of customer because before creating a product, a core understanding or knowledge of the product is very necessary.

Planning:-
Once we are clear with the project requirements, they are properly documented. An estimated scheduling process of the projects has been designed. This phase describes how technical tasks are going to take place and what resources are needed and how to use them.

Modelling:-
This involves specifying and designing the hardware and software requirements of the project, and their inter-relation. The entire software of the project is broken down into different logical modules or blocks which are identified and systematically documented.

Construction:-
This is the next step which involves nothing but writing software code and actually implementing the programming ideas and algorithms which have been designed or decided upon in the 'Modelling' phase. Once the coding and implementing phase has been completed, it is now time to test the code. The code that has been written is subjected to a series of tests and test cases to detect and determine whether there are any bugs, errors or software failures.

Deployment:-
Once all the repair work, i.e. correcting and re-writing every piece of erroneous or flawed code is completed, we then move to the next and the last phase titled 'System Deployment and Maintenance'. As the name suggests, the last phase is nothing but handling over the completed project to the customer, and subsequently performing maintenance activities on a period basis.

Fly with ease Increments:-

- The incremental model delivers series of releases to the customer that release are called increments more and more functionality is associated with each release or with each increment.
- The first increment is called core product, in this release the basic requirements are implemented and then in subsequent increment new requirement are added. Mainly our core product of this project contains Order Module, Payment Module, Messaging feature and flight details module.
    - In our application our first build will be released to the customer after the time interval of 2 months and 1 week.
- The second increment of this project Contains Weather module, Entertainment section and Review module.
    - Later on as estimation our second build will be release after 1 month and 3 weeks.
- Further More releases will be depend on customer's requirements.

SOFTWARE REQUIREMENTS SPECIFICATIONS

Introduction

This section gives a scope description and overview of everything included in the SRS document. Also, the purpose of this document is described.

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the "Fly with Ease" software. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints interface and interaction with other applications. This document is primarily intended to be purposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 Scope

The "Fly with Ease" software is an interactive seatback screen that help in providing a comfortable and luxurious travelling experience to the flyers. It helps to reduce the communication gap among the flyers, air hostesses and stewards. It provides an array of entertainment (like music, movies, games etc.), food order facility, and information regarding to flight and destination as well.
This will be an Inflight Entertainment System consists of administrator (for database, security and information) and the end users (flyers, cabin crew like air hostesses and stewards).
Actors would register themselves, so that, the screen will display the content regarding to that particular actor and all can connect with each other in the network.

1.3 Definitions, Acronyms and Abbreviations

Definitions:

- Flyers: Passengers of the flight called Flyers.
- Flight Attendants: it is the member of aircrew employed by airlines aboard commercial flights, primarily to ensure the safety and comfort of the flyers.
- Air Hostess: female Flight Attendant known as Air Hostess.
- Steward: male Flight Attendant known as Steward.
- Router: This is a hardware device that routes data from a local area network (LAN) to another network connections.
- Victual: It can be anything that can be served or used as a food.

Acronyms:

GUI: Graphical User Interface
IFE: Inflight Entertainment
SRS: Software Requirement Specification
GPS: Geographic Positioning System
XML: Extensible Mark-up Language
TCP/IP: Transfer Control Protocol/Internet Protocol
Wi-Fi: Wireless Fidelity

Abbreviations:
i.e.: that is
etc.: etcetera

1.4 References

- "Software Engineering- A practitioner's approach (7th edition)" by Roger S. Pressman.
- "Software Engineering (Revised 2nd edition)", by K.K. Aggarwal & Yogesh Singh.
- 830-1998-IEEE recommended Practice for Software Requirement Specifications.

1.5 Overview
The rest of the SRS document describes various system requirements, interfaces, features and functionalities in detail.

2. The Overall Description
This section will give an overview of the whole system. This system will be explained in its context to show how the system interacts with other systems and introduce the basic functionalities of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last the constraints, all assumptions for the system will be presented.

2.1 Product Perspective
This software is a window based, self-contained and an independent system. This system involves two end users i.e. flyers and flight attendants and an administrator. The database connections is provided offline from airline servers for fastest access to the data.

2.1.1 User Interface
The user interface for the software shall be compatible to run on windows by which user can access it. It is a GUI system. It will be touch screen and given on the seatbacks. This application will be installed at the seatbacks.

- A login screen will be provided.
- A screen to choose their native languages.
- A screen where flyer can view all the sections like entertainment hub, weather prediction, victuals ordering section and flight details.
- Flyers can search for a particular movie, songs and games.
- Flyers can have the weather details from the weather prediction section.
- Admin can add latest entertainment stuff via upload module.
- Flight Attendants view order on their screen.

2.1.2 Hardware Interface
Hardware required to provide wireless system for allowing the interaction between each seat in the aircraft and flight attendants. Along with a complete intranet to deal with, software must be reliable when communicating to and from the main processor.

2.1.3 Software Interface

- FWE shall communicate with administrator to upload the database for the end-users to login therefore a database management system will required.
- FWE system will provide interface to fetch the data related to their entries.

2.1.4 Communication Interface

The intranet communication will be through TCP/IP protocol suite. Airplane connects to satellites which sends and receive signals to earth via receivers and transmitters. Wi-Fi signal is distributed via on a board router.

2.2 Product Function

- User must have a valid user I'd and password.
- Administrator regular check the database of users and upload into the software for creating a new user account.
- After logging in user can access any option he/she want.
- Flyer can watch the entertainment stuffs.
- Flyers can add the items into their favourite details (videos, movies, songs and games).
- In victuals section order will place and record will be maintained through the database.
- Admin will add and delete the movies, music, news etc.
- Flyers can rate the software and submit their reviews.
- Admin can view the record of orders and reviews given by flyers.
- Flyers can view the weather details by updating their location.
- Flyers can view the flight details.

2.3 User Characteristics

- The end users must have the basic knowledge about computer and internet, so that they can use this application.
- The administrators must be well versed with database management system.

2.4 Constraints

- Only admin can upload and delete the entertainment stuff.
- There can be only one admin.

3. Specific Requirements

3.1 External Interfaces:

3.1.1 User Interface:

Since the software is particularly installed by flights agencies for their flyers on the screen placed in front of their seats, so the screen will be worked as user interface for the software. The user interface for the software shall be compatible to use satellite by which weather information can be update time to time.

3.1.2 Hardware Interface:
Since the application must run over the intranet, all the hardware shall require to connect intranet will be hardware interface for the system. As for e.g., Modem, WAN-LAN, Ethernet Cross-cable.

3.1.3 Software Interface:
Operating System: Windows 10
Dev-C++ (for coding/developing the software)
My Sql as DBMS for database

3.1.4 Communication Interface:
The system shall use the TCP/IP protocol suite for the intranet communication and the system shall also use the HTTP protocol for communication over the internet.

3.2 Functions:
The software provides the following functionalities:

- **Register Module:**
  In this module, all the actors can create their new account.

- **Login/Logout:**
  Actors have to enter their username and password to login their account accordingly and they can also logout their account from the software.

- **Select Language:**
  This module will give the opportunity to user to select their native language among several languages.

Functionalities (Only For Flyers)

- **Entertainment Section:**
  This module is for flyers entertainment and it contains a huge collection of movies, songs, and games. User can perform following functionalities in this module:-
  - Watch movies
  - Listen music
  - Play games
  - Search: - User can search the movies, songs and music by their name.
  - Add to Favourite: - User can add their favourite movies and songs in their favourite list so that they do not have to search them again and again.

- **Victuals Section:**
  In this flyer can place their order from the menu items.
  - Place Order: - User can place their order by just adding the items to the cart.
  - Pay Bill: - User can pay for their bills by entering the details of their debit or credit card and for authentication they received unique 4-digit otp code through sms.

- **Weather Prediction Section:**
  The primary objective of this module is to show the prevailing weather conditions hourly or even minute-to-minute by using location services to the flyers.

- **View Details:**
  User can view all the details of flight like departure time, number of stops etc.

- **Acquaint Requirements:**
  Flyers can message the air hostesses for their other needs like towels, blankets etc.

- **Review:**
  Flyers can rate the software and submit their comments in this section.

Functionalities (Only For Administrator)

- **Add Movies, songs and games:**
  This section is available only for the admin so that they can upload the latest movies, songs and games in the software, so that flyers can view.

- **Delete Movies, songs and games:**
  This section is available only for the admin so that they can delete the old movies, songs and games from the software

- **View Record of Orders:**
  In this section administrator can view all placed orders.

- **View Record of Reviews:**
  In this section administrator can view all the ratings and reviews given by passengers.

Functionalities (Only For Flight Attendants)

- **View Order:**
  This section is available for the air hostesses and for stewards so that they can view placed order given by flyers.

- **View Messages:**
  In this section, flight attendants can view the messages given by passengers and can accept the order of their miscellaneous items.

3.3 Performance Requirements:-

FWE shall be available 24 hours a day, 7 days a week.
The software shall requires the internet for updating the weather and paying bill.
Response time of the "Fly with Ease" should be less than 10 second most of the time. Response time refers to the waiting time while the system accesses, queries and retrieves the information from the databases.
FWE shall be able to handle at least 500 transactions or inquiries per second.
FWE shall be robust enough to have a high degree of fault tolerance. For example, if the user enters the wrong password or the no. of users at a time at the software is too large the system should not crash and shall produce a suitable error message.
FWE shall be alleviate to use as it is just similar to a touch screen of a cell phone. It will provide an environmental interaction between the actors of the software.
The software shall take initial load time at the time when admin have to upload the movies and songs.

3.4 Logical Database Requirement:-

FWE software based on three databases i.e., Payment Database which store the information of all the payment details, Whether info database which stores the details of the weather and a local server database which contains the number of users on the software, so that the user can login safely. The similar database contain information about the victuals availability and the entertainment section too but there will be the different tables containing different records. Only system administrator has the right to change system parameters in the databases. The system should be secure and must use encryption to protect the database.

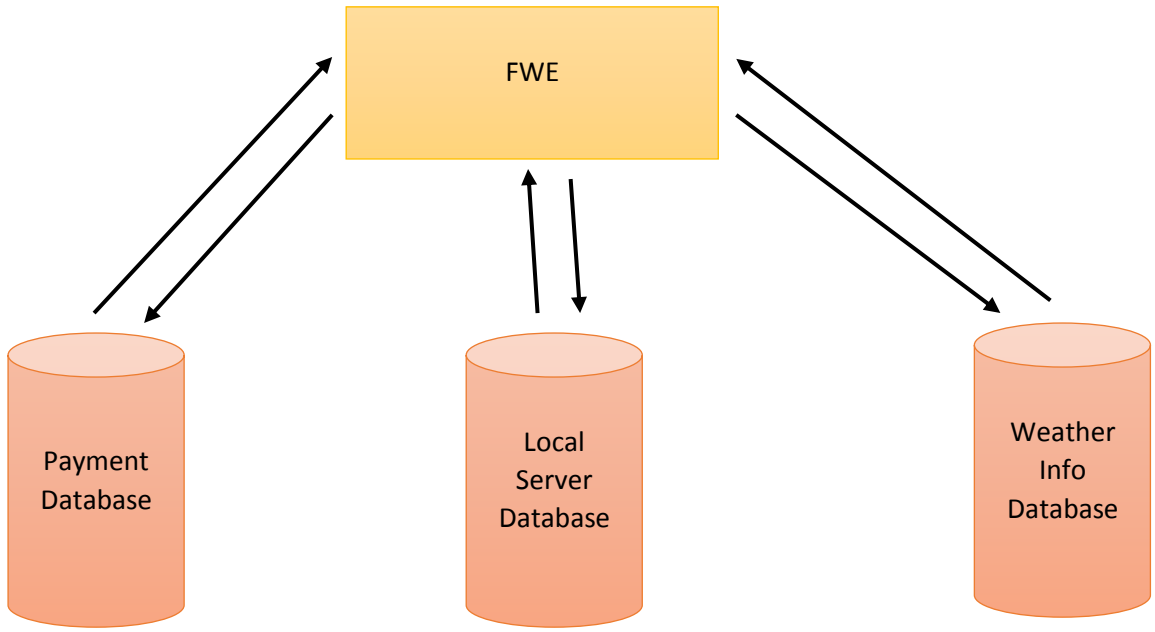


Fig 2: Logical Database

3.5 Design Constraints:-

- A centralized database system is required.
- There will be a memory requirement.
- The product must be stored in such a way that it is easily available to the user.
- Regular updating is necessary by the admin.

3.6 Software System Attributes:-

3.6.1 Safety and security considerations:-

The application must be existed always normally.
It must be ensured that access will be provide to the authorized person through user id and password.
Checks will be performed by the admin at the regular intervals to ensure data integrity.
The system should resist accidental or deliberate intrusions, when users operate on the system If the system should not resist, the important information or data – such as the credit card number, id number, username etc.-which belongs to the user shall be stolen by the hacker .So security of the system is very important for the user.

3.6.2 Reliability:-

- The FWE shall not fail on average more than once per week.
- The FWE shall not take on average longer than two days to repair after a system failure.

3.6.3 Availability:

The system shall be available 95% of the time (24 hours of the 7 days) unless previously announced for scheduled maintenance or backup.

3.6.4 Maintainability:-

In the event that the system will undergo for a known system maintenance or data backup or under updations, users of the system shall be notified during next interaction with the system.
The source code developed by this system shall be maintained in configuration management tool.

3.6.5 Portability:-

The system shall allow all information and applications to be run from that particular screen available at the passenger seat and crew members or admin have.

3.7 Other Requirements:

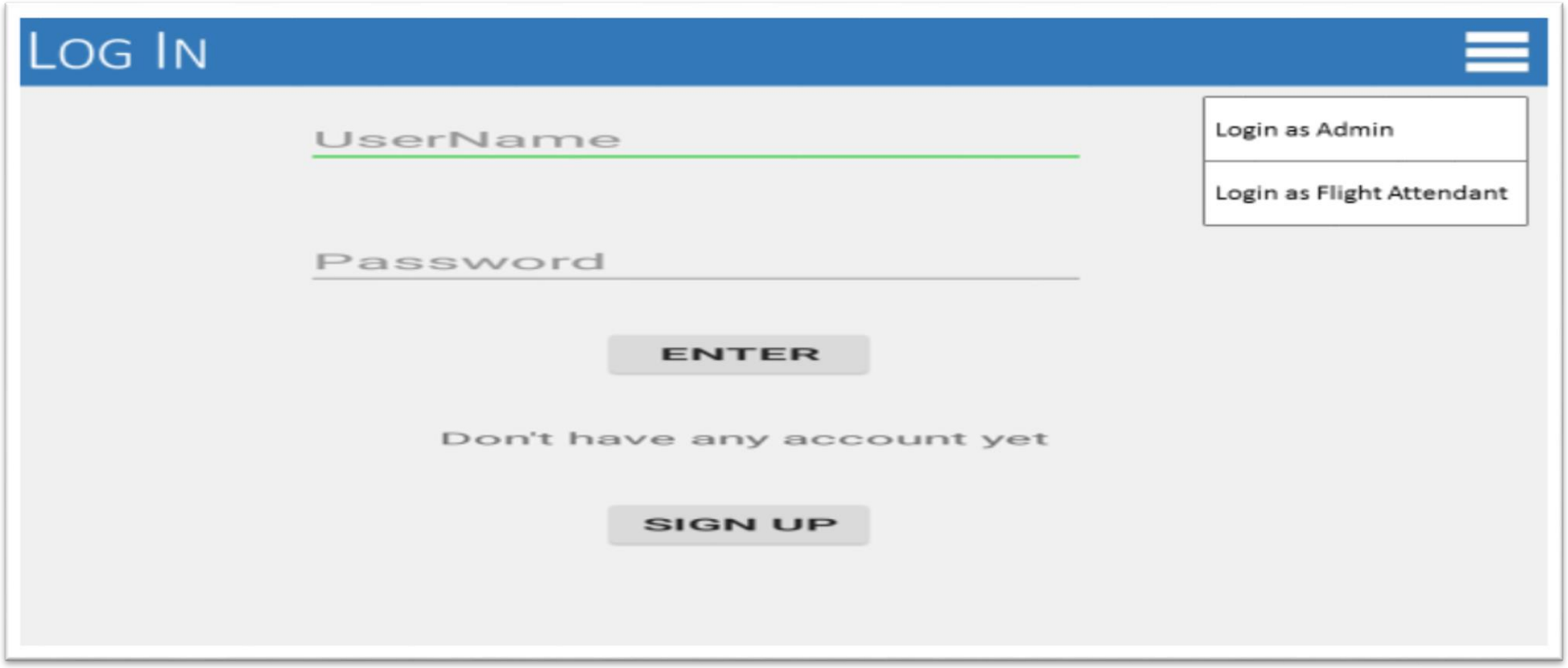
4. Change management process:-

The system must allow addition feature to integrate in the system in the near future.

Screens

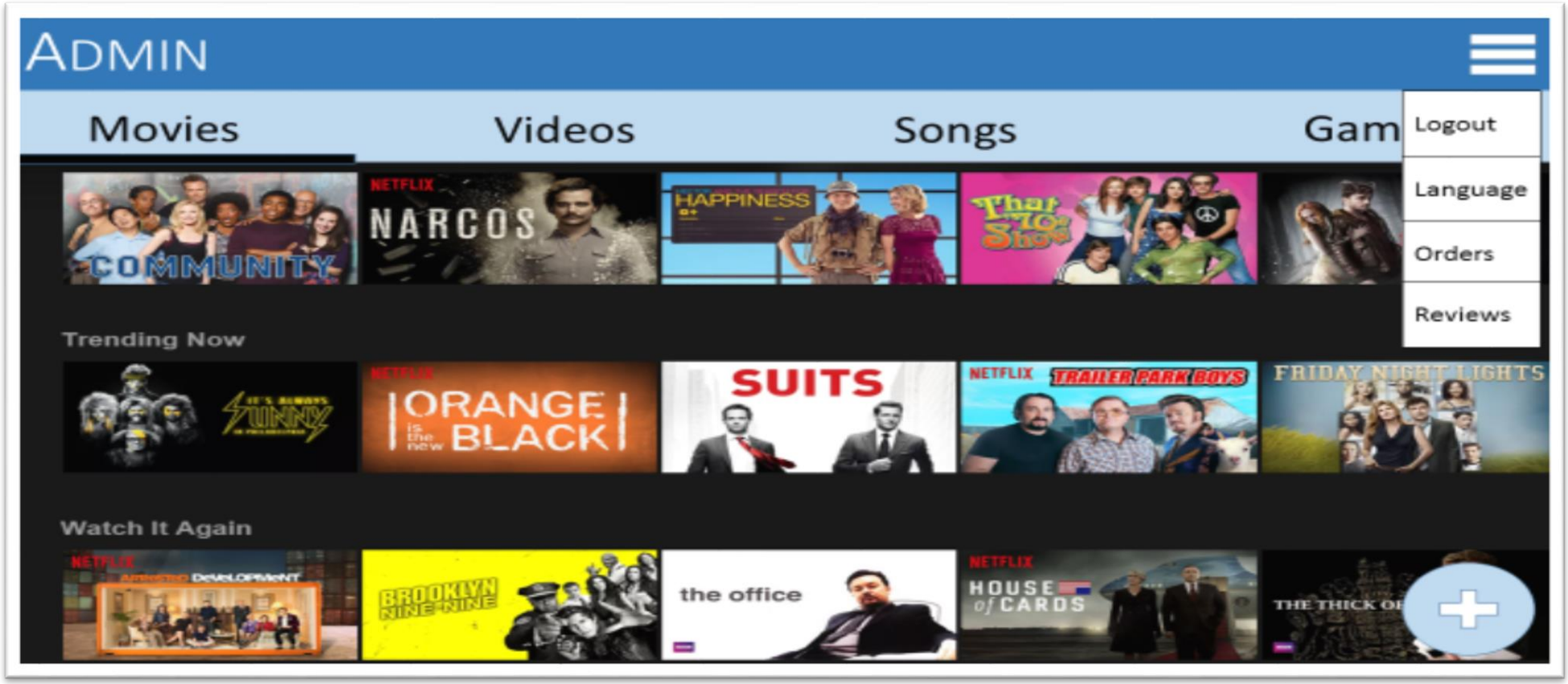
Number of external inputs = 0
Number of external outputs = 0
Number of external inquiries = 0
Number of internal logical files = 0
Number of external interface files = 0

Number of external inputs = 1
Number of external outputs = 0
Number of external inquiries = 0
Number of internal logical files = 1
Number of external interface files = 0

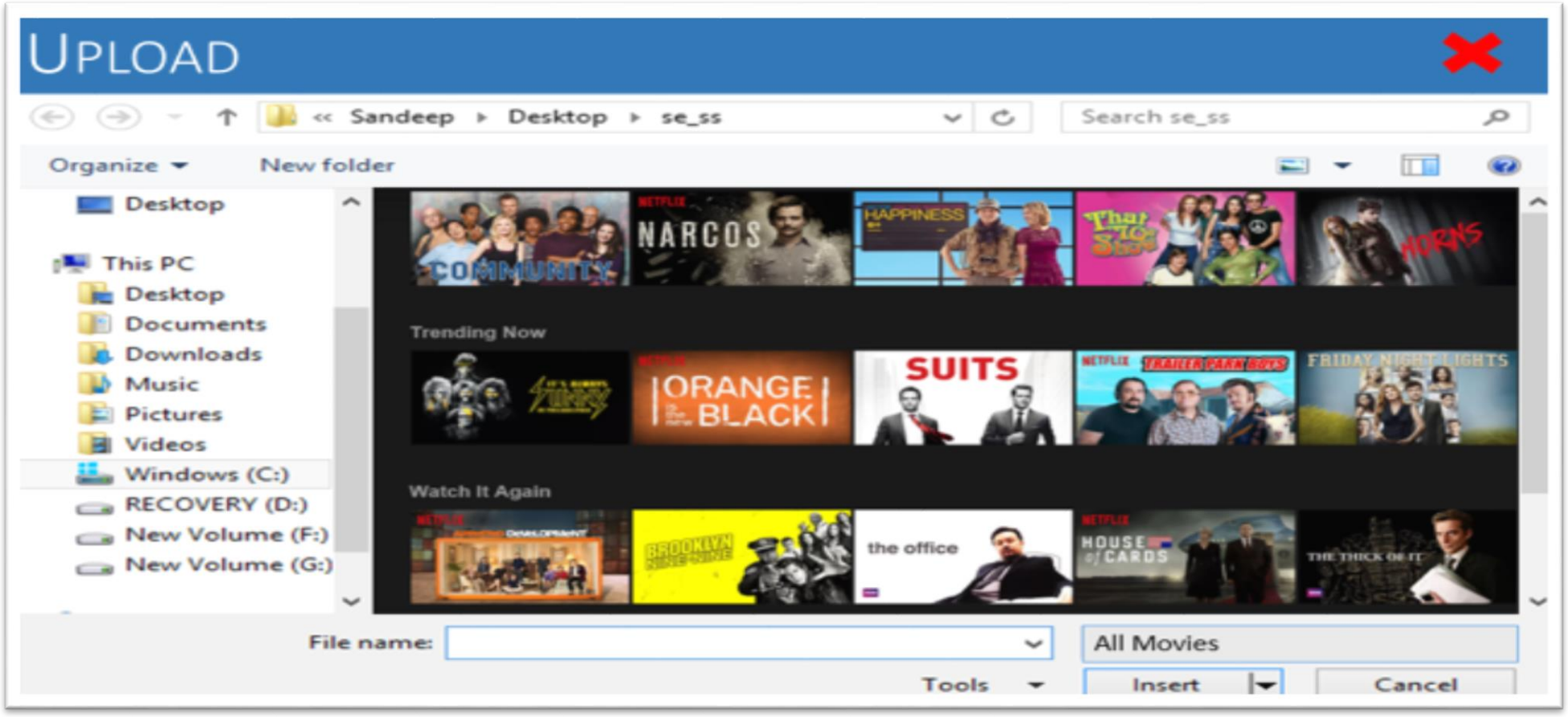


Number of external inputs = 2
Number of external outputs = 1
Number of external inquiries = 0
Number of internal logical files = 1
Number of external interface files = 0

Number of external inputs = 5
Number of external outputs = 1
Number of external inquiries = 0
Number of internal logical files = 1
Number of external interface files = 0



Number of external inputs = 1
Number of external outputs = 1
Number of external inquiries = 0
Number of internal logical files = 1
Number of external interface files = 0



Number of external inputs = 1
Number of external outputs = 1
Number of external inquiries = 1
Number of internal logical files = 1
Number of external interface files = 0

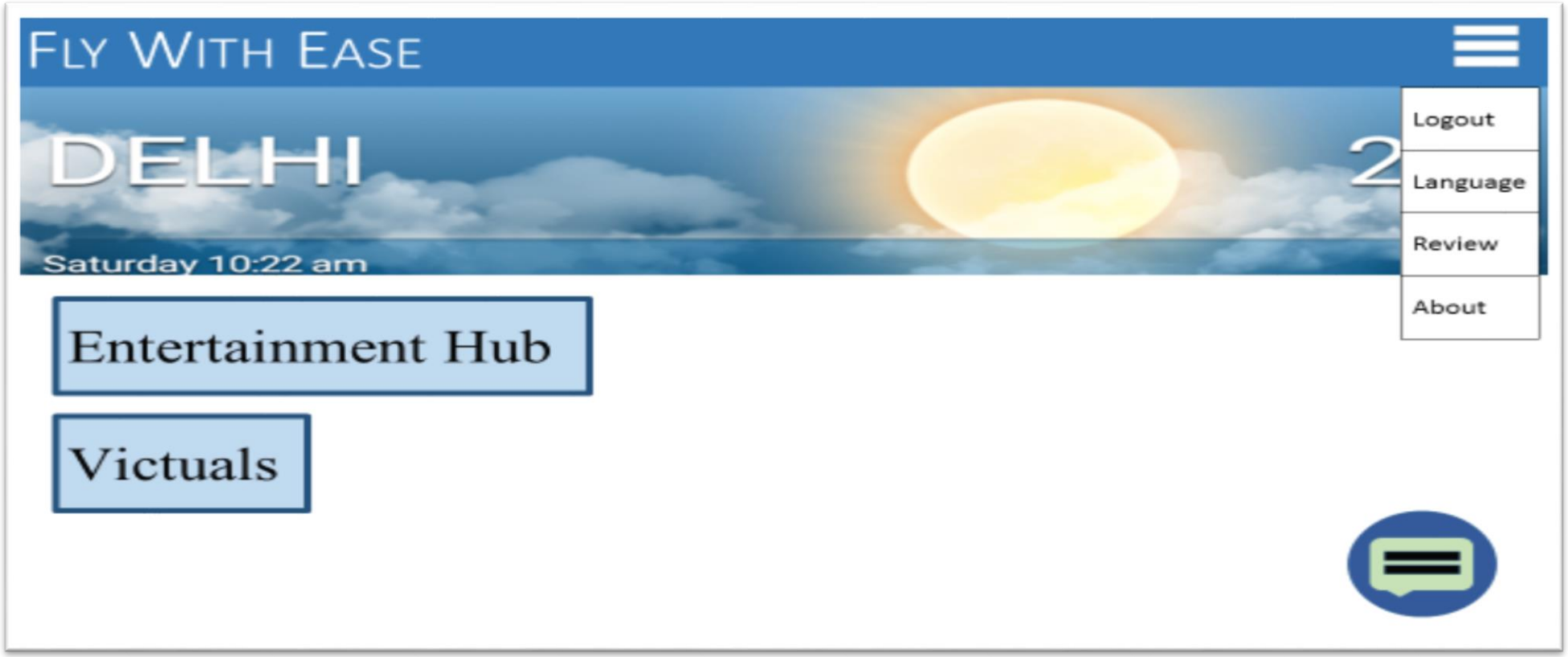
Number of external inputs = 1

Number of external outputs = 0

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 0



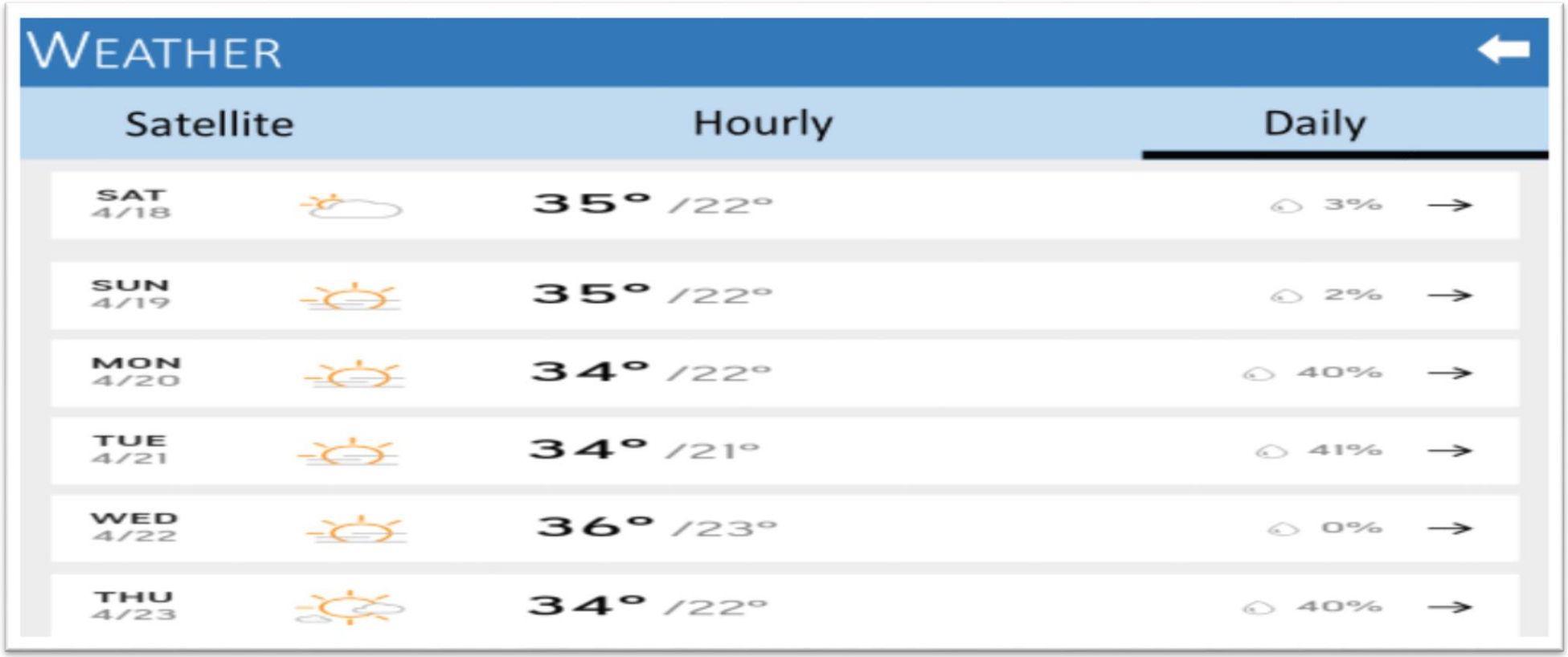
Number of external inputs = 1

Number of external outputs = 0

Number of external inquiries = 0

Number of internal logical files = 4

Number of external interface files = 1



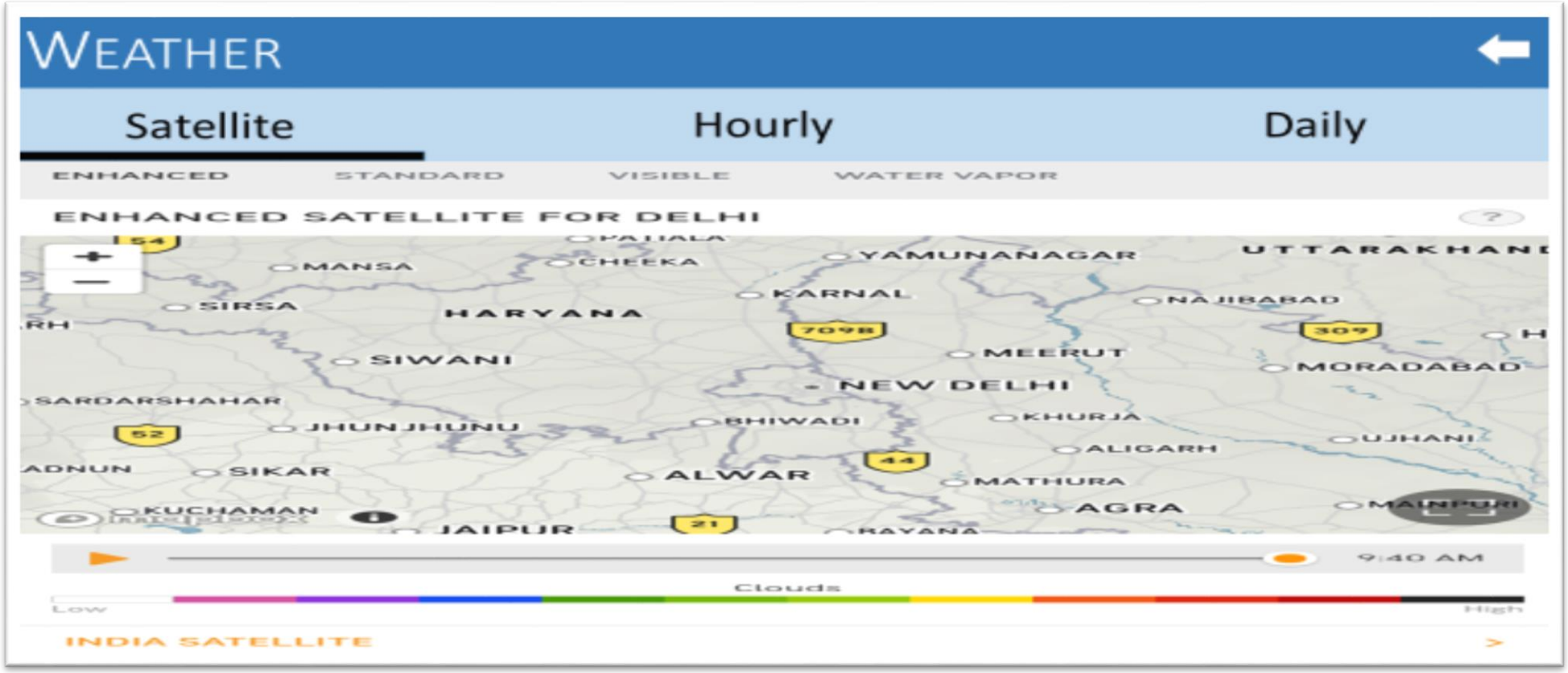
Number of external inputs = 0

Number of external outputs = 0

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 1



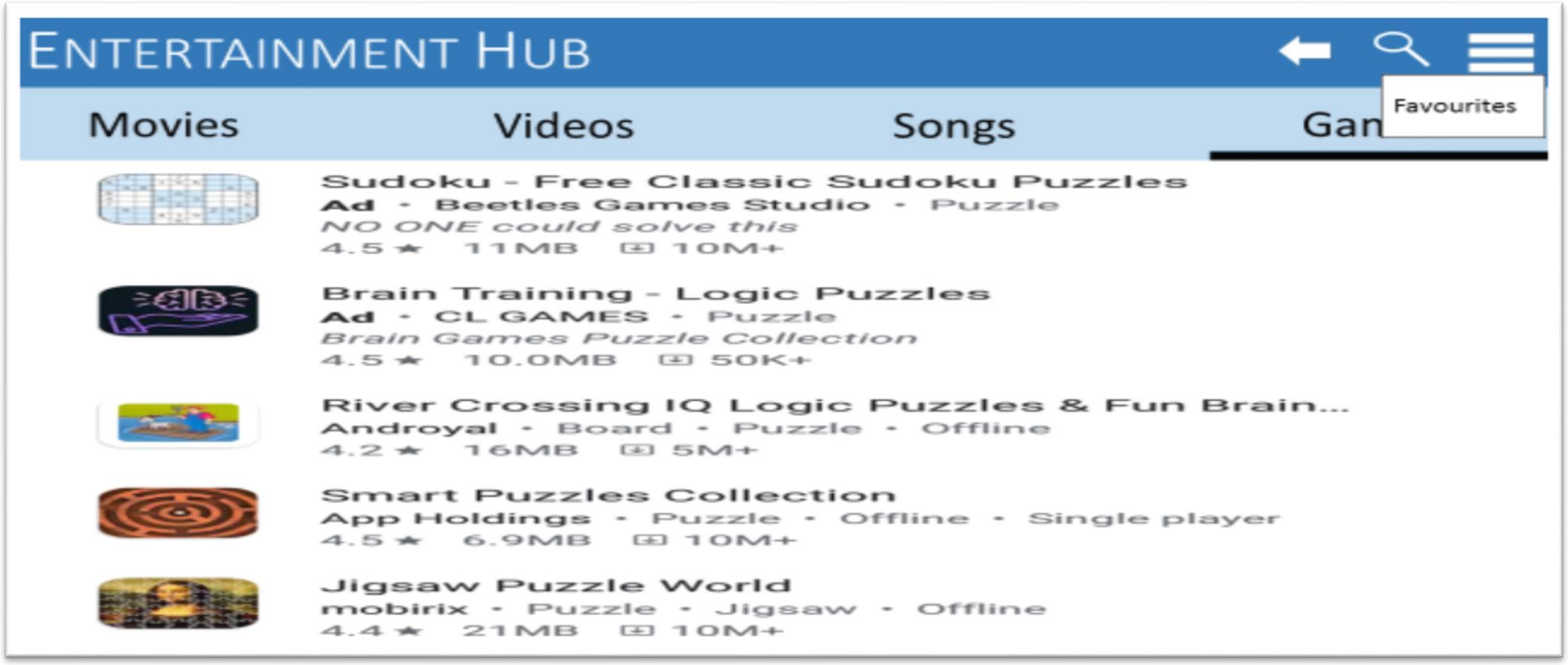
Number of external inputs = 1

Number of external outputs = 1

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 1



Number of external inputs = 1

Number of external outputs = 1

Number of external inquiries = 1

Number of internal logical files = 1

Number of external interface files = 0

Number of external inputs = 1

Number of external outputs = 1

Number of external inquiries = 1

Number of internal logical files = 1

Number of external interface files = 0

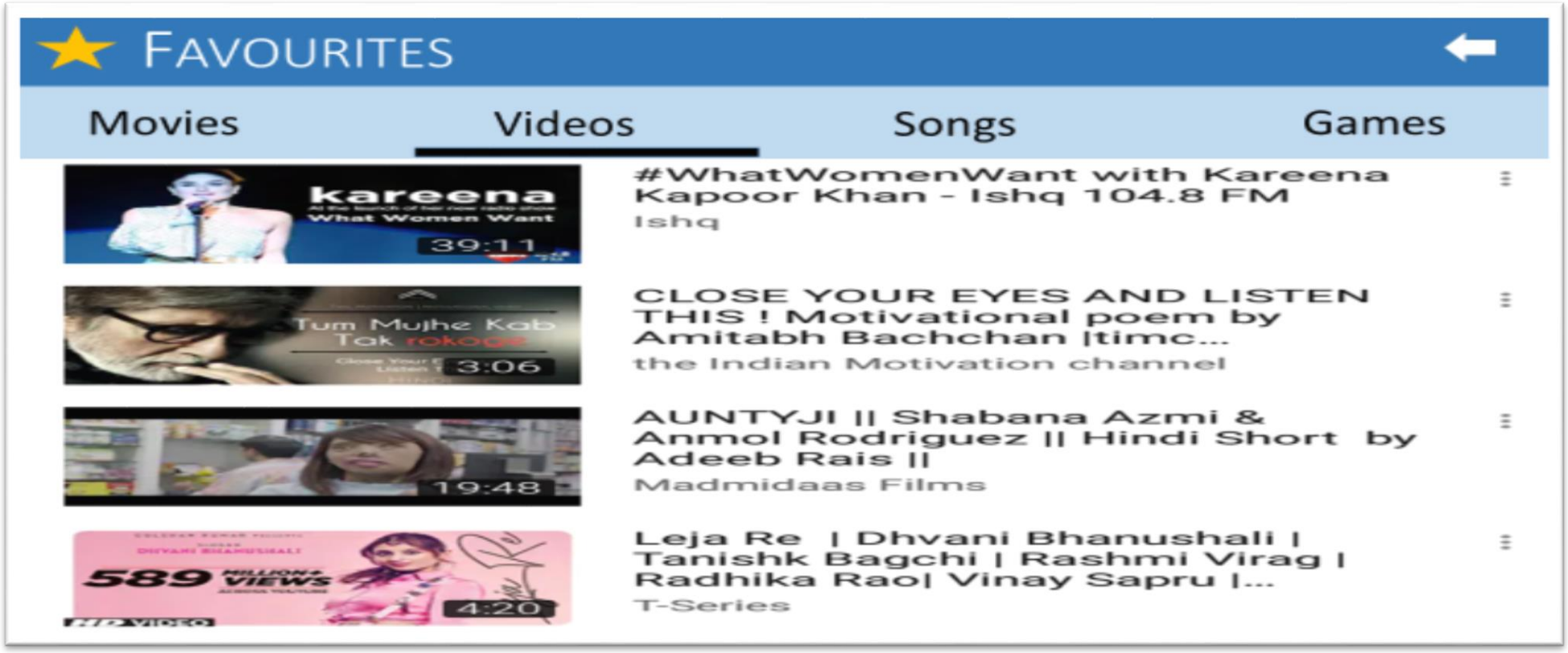
Number of external inputs = 0

Number of external outputs = 1

Number of external inquiries = 1

Number of internal logical files = 1

Number of external interface files = 0



Number of external inputs = 1

Number of external outputs = 1

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 0

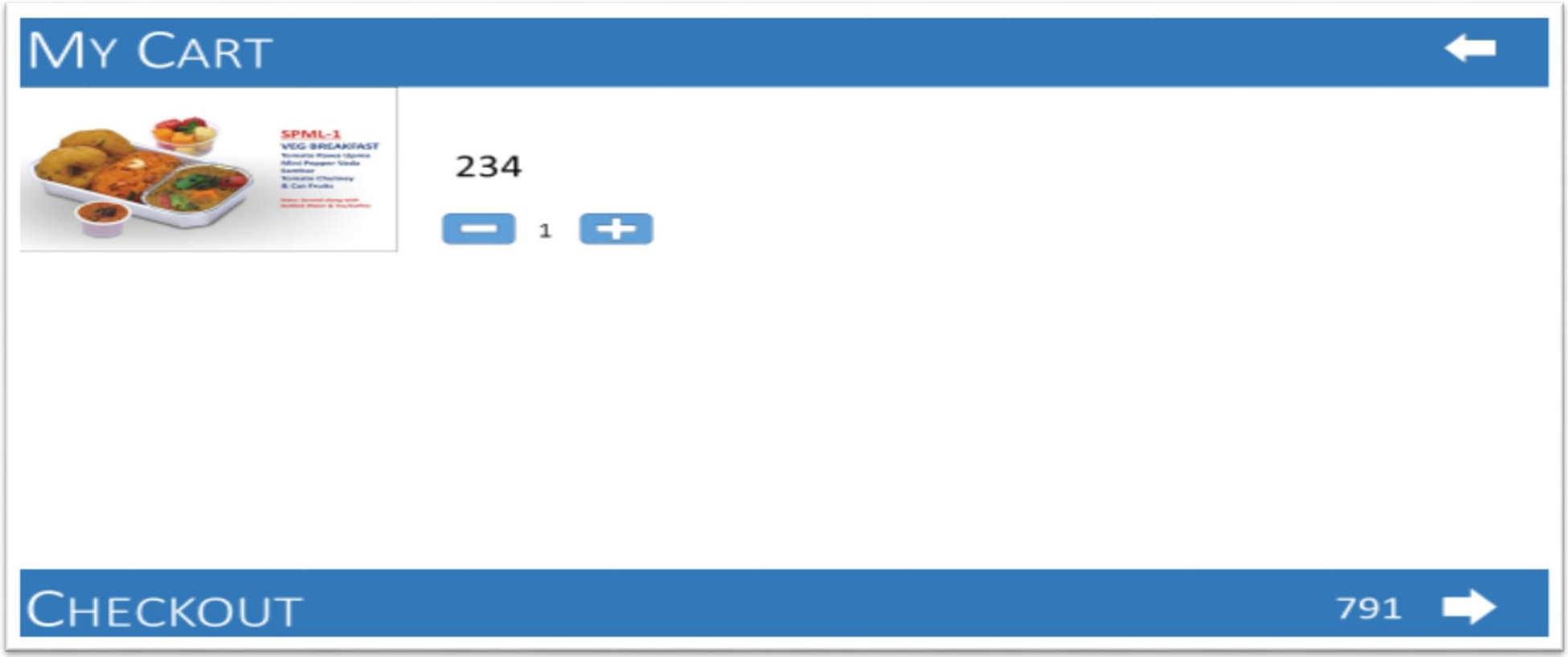
Number of external inputs = 15

Number of external outputs = 1

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 0



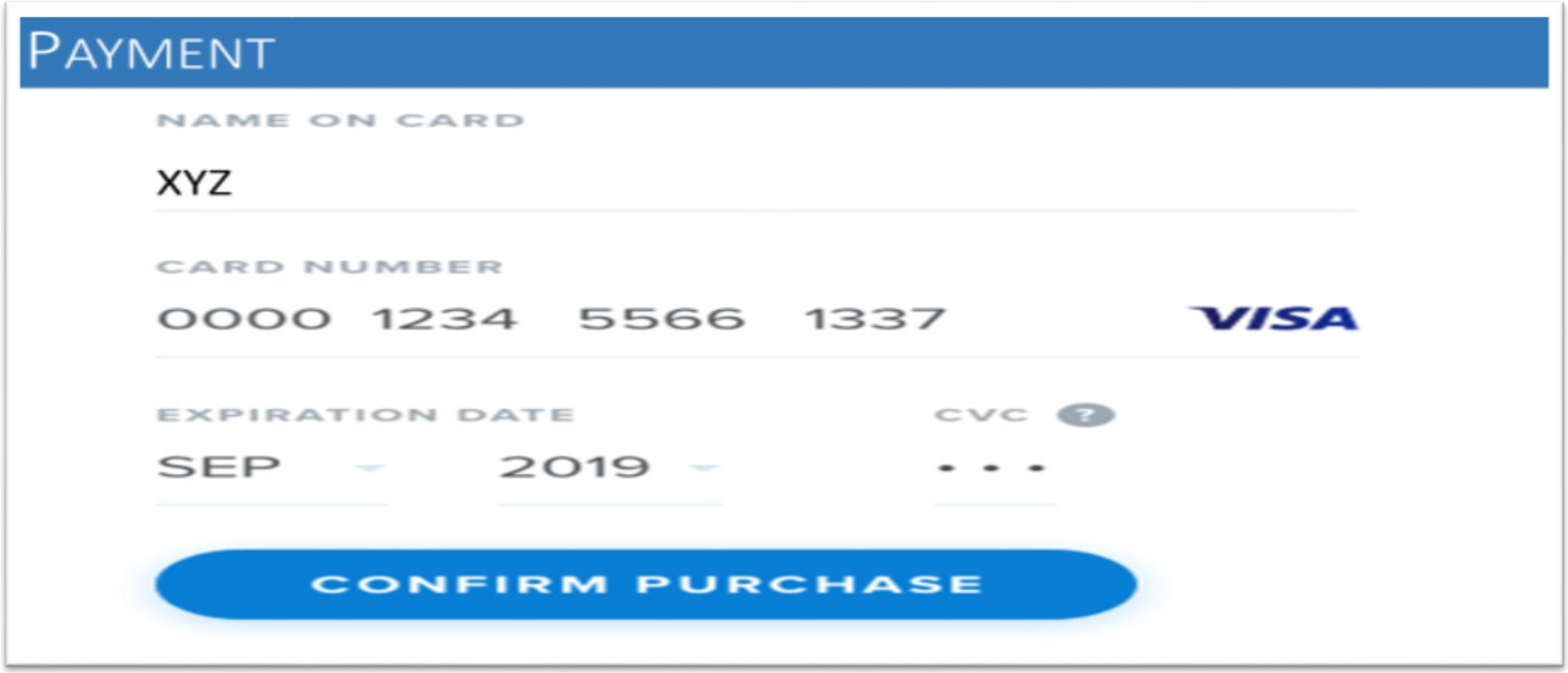
Number of external inputs = 1

Number of external outputs = 1

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 0



Number of external inputs = 5

Number of external outputs = 1

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 1

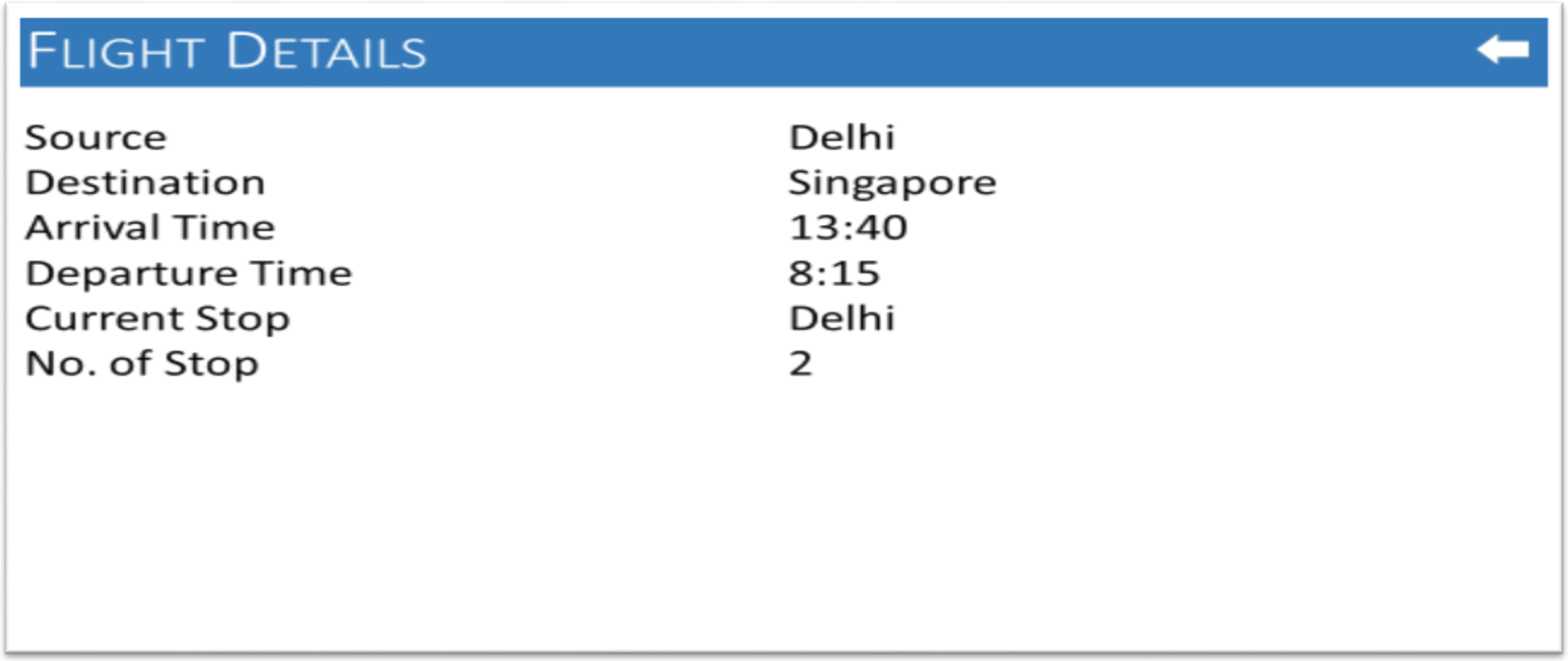
Number of external inputs = 0

Number of external outputs = 0

Number of external inquiries = 0

Number of internal logical files = 0

Number of external interface files = 0



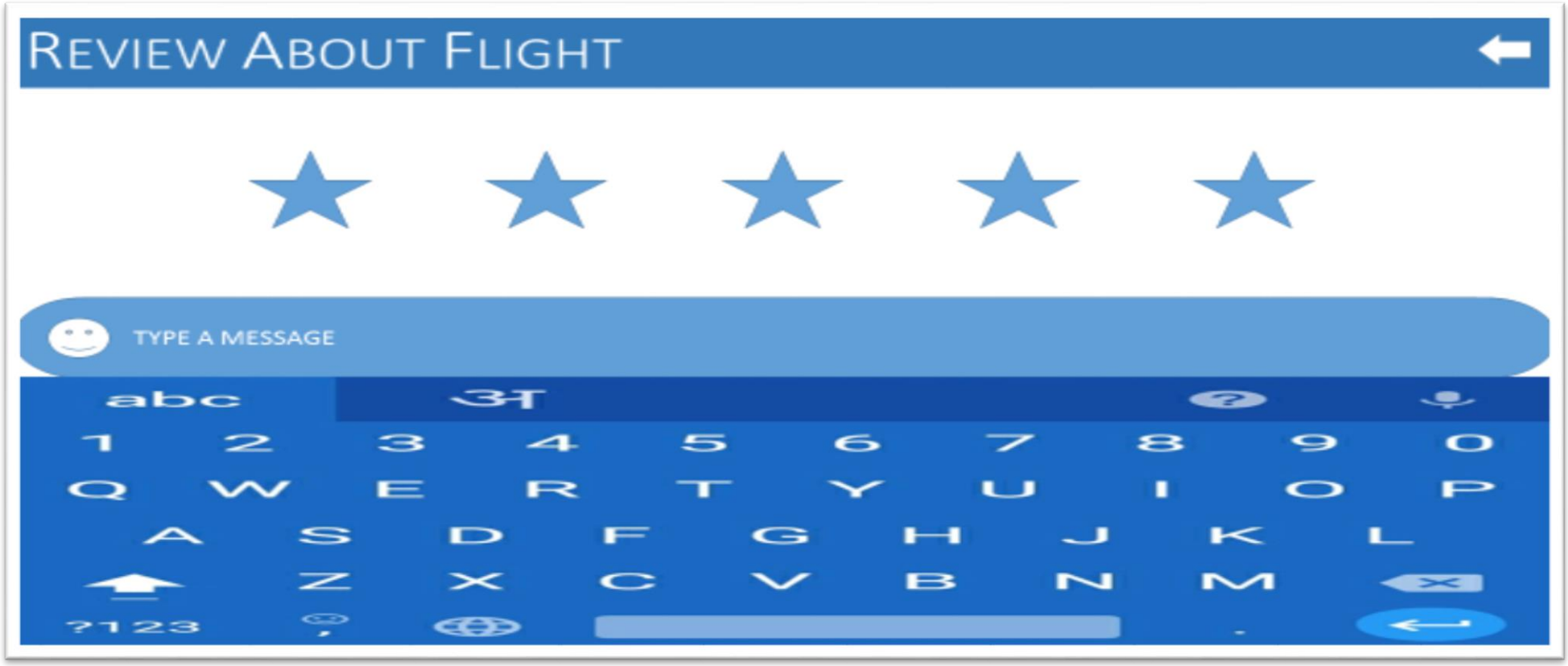
Number of external inputs = 0

Number of external outputs = 0

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 0



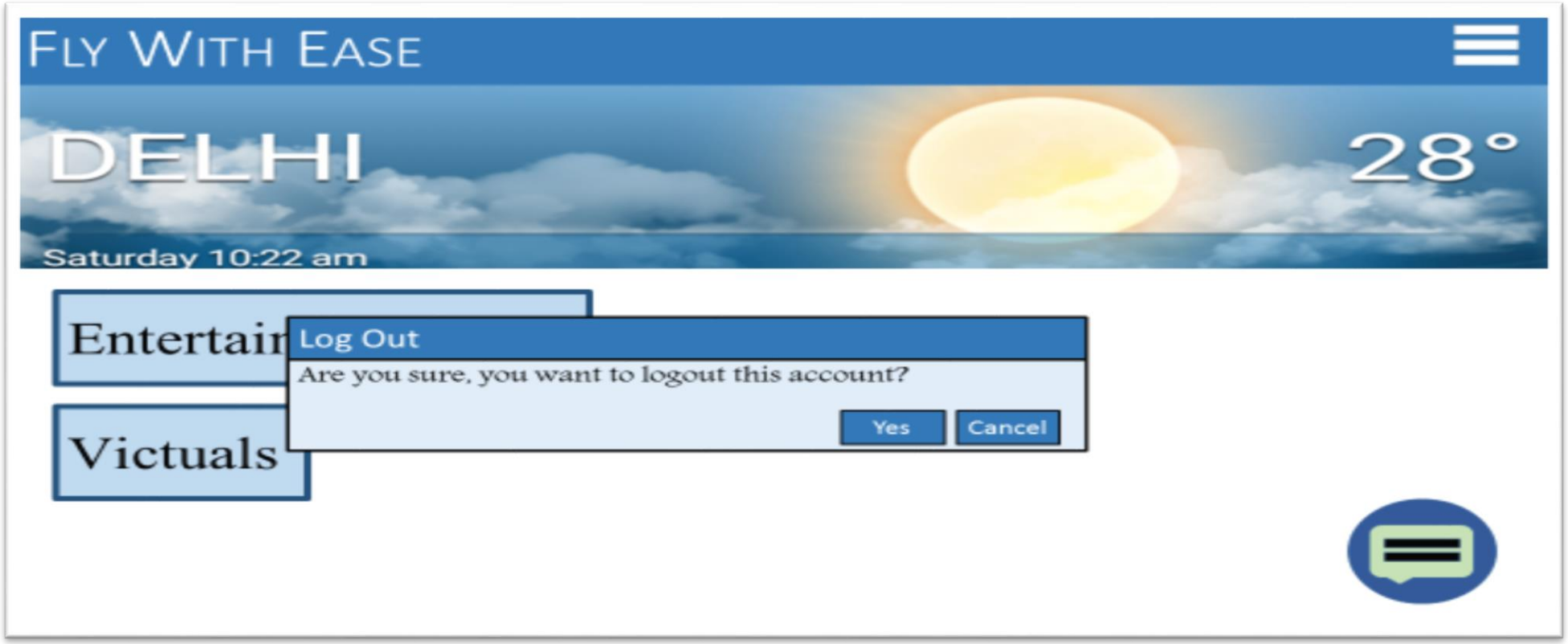
Number of external inputs = 2

Number of external outputs = 1

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 0



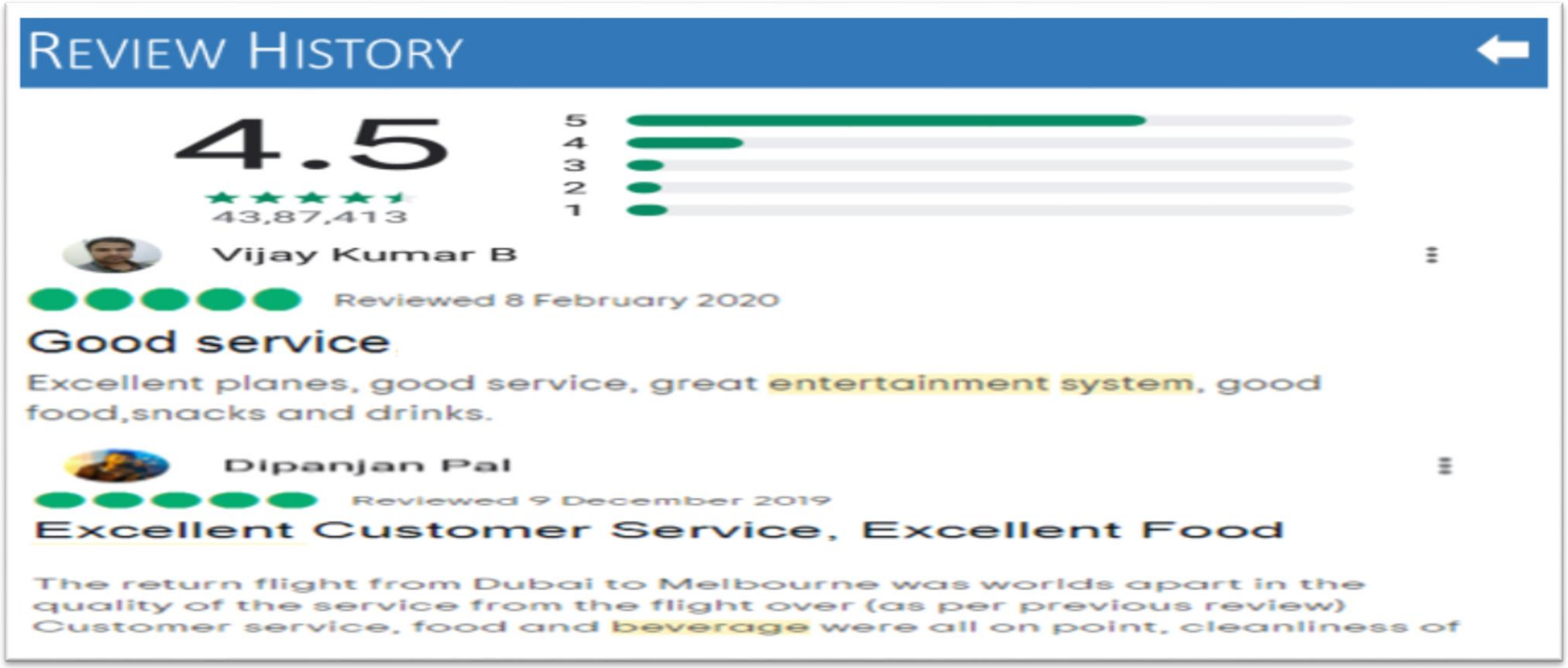
Number of external inputs = 1

Number of external outputs = 0

Number of external inquiries = 0

Number of internal logical files = 0

Number of external interface files = 0



Number of external inputs = 0

Number of external outputs = 0

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 0

ORDER HISTORY

24th February 2020

Order No.	Seat No.	Time	Dish Code	No-of-dishes	Price	Bill
101	R-3,A	14:17	SPML-12	1	189	189
102	R-10,D	14:56	SPML-3	1	324	324
103	R-15,B	15:07	SPML-7	2	357	714
104	R-4,C	16:05	SPML-2	1	584	584
105	R-4,C	16:05	SPML-7	1	357	357
106	R-7,F	17:34	SPML-9	2	67	134

25th February 2020

Number of external inputs = 0

Number of external outputs = 0

Number of external inquiries = 0

Number of internal logical files = 1

Number of external interface files = 0

Project Table

Work Task	Planned Start	Actual start	Planned Complete	Actual Complete	Assigned Person	Effort Allocated
Problem Statement	M1,W1	M1,W1	M1,W2	M1,W2	Vaishnavi, Sakshi	4 P-W
Software model	M1,W2	M1,W2	M1,W3	M1,W3	Anjali, Vaishnavi	4 P-W
SRS	M1,W3	M1,W3	M1,W4	M1,W4	Vaishnavi, Anjali	4 P-W
ERD	M1,W4	M2,W1	M2,W1	M2,W1	Sakshi, Anjali	2 P-W
Data Dictionary	M2,W1	M2,W1	M2,W2	M2,W2	Sakshi, Anjali	4 P-W
Context level Diagram	M2,W2	M2,W2	M2,W3	M2,W3	Sakshi, Vaishnavi	4 P-W
DFD Level 1	M2,W3	M2,W3	M2,W3	M2,W3	Sakshi, Vaishnavi, Anjali	6 P-W
DFD Level 2	M2,W3	M2,W3	M2,W4	M2,W4	Sakshi	2 P-W
Use case Diagram	M3,W1	M3,W1	M3,W1	M3,W1	Vaishnavi	1 P-W
Use case Description	M3,W1	M3,W1	M3,W2	M3,W2	Anjali	2 P-W
Function-Point Matrices	M3,W2	M3,W2	M3,W3	M3,W3	Vaishnavi	2 P-W
Risk analysis	M3,W3	M3,W3	M3,W4	M3,W4	Vaishnavi, Sakshi, Anjali	6 P-W
COCOMO|| model	M3,W4	M3,W4	M4,W1	M4,W1	Anjali	2 P-W
Source Code	M4,W1	M4,W1	M4,W2	M4,W2	Sakshi	2 P-W
Testing	M4,W2	M4,W2	M4,W3	M4,W4	Vaishnavi, Sakshi, Anjali	6 P-W

TIME LINE CHART

WORK TASK	JANUARY				FEBRUARY				MARCH				APRIL			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Problem Statement																
Software Model																
SRS																
ERD																
Data Dictionary																
Context level Diagram																
Dfd level 1																
Dfd level 2																
Use case diagram																
Use case description																
Function Point Metrices																
Risk Analysis																
Cocomo model II																
Source code																
Testing																

EFFORT ESTIMATION USING COCOMO II MODEL

COCOMO, Constructive Cost Model is the most widely used software cost estimation models in the industry. It has evolved into a more comprehensive estimation model called **COCOMO II. COCOMO II** is actually a hierarchy of estimation model that addresses the following areas:

- **Application Composition Model**: This model is used during the early stages of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity and paramount.
- **Early Design Stage Model**: This model is used when requirements have been stabilized and basic software architecture has been established.
- **Post Architecture Stage Model**: This model is used during the construction of the software.

The COCOMO II model requires sizing information. These different sizing options are available as part of the model hierarchy:

- Object points
- Function points
- Lines of source code

The COCOMO II application composition model uses object points. Object point is an indirect software measure that is computed using counts of the following:

- No. of screens (at the user interface)
- No. of reports
- No. of components required to build the application

Each object instance (e.g. a screen or report) is classified into one of the three complexity levels (i.e. simple, medium or difficult). Complexity is a function of the number and source of the client and server data tables that are required to generate the screen or report and the no. of views or sections presented as part of the screen or report.

COMPLEXITY WEIGHTING FOR OBJECT TYPES

Object Type	Complexity Weight		
	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
3 GL Component			10

Fig 3: Complexity weighting for object types

The object point count is determined by multiplying the original number of the object instance by the weighting factor and summing to obtain a total object point count.
When component-based development or general software reuse is to be applied, the percent of re-use is estimated and count is adjusted.

Since, we are not using any of the components, the percent of reuse (% reuse) here is zero.
NOP = (Object point) x [(100- % reuse) / 100]
Where NOP is defined as new object points.
To derive an estimate of effort based on the computed NOP value, a "productive rate" must be derived as:

PROD = NOP/ person-month
For different levels of developer experience and development environment maturity.
Once the productivity rate has been determined an estimate of project effort is computed using:
Estimated Effort = NOP / PROD

Productivity Rate for Object Points:

Developer's Experience/ Capability	Very Low	Low	Nominal	High	Very High
Environment Maturity/ Capability	Very Low	Low	Nominal	High	Very High
PROD	4	7	13	25	50

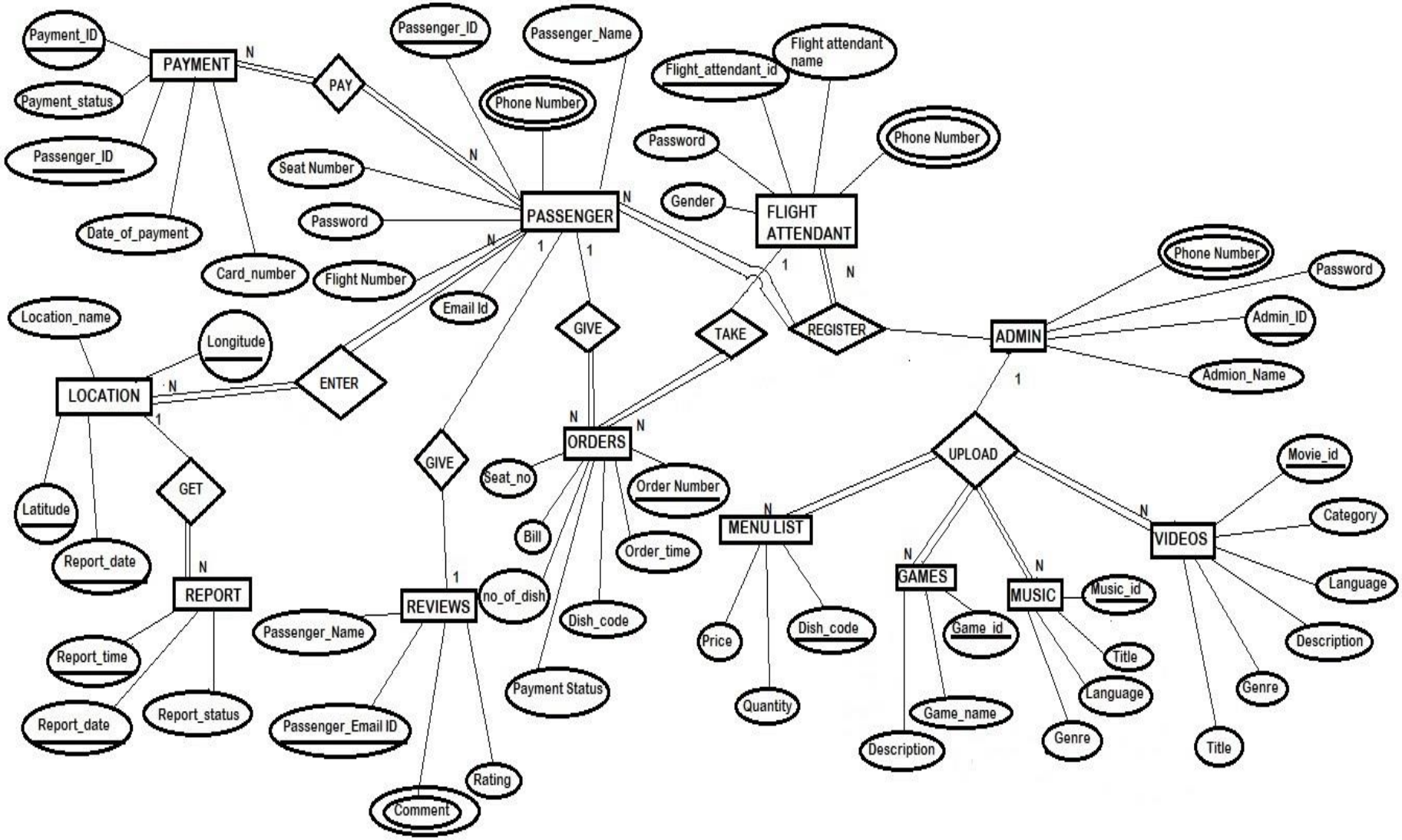
COCOMO Estimation of Our Project

No. of screens =19 (simple=7, medium=10, Difficult=2)
No. of reports = 4 (simple=2, medium=2)
No. of reusable components=0

Object Points:

For screens= (7*1) + (10*2) + (2*3) = 33
For reports= (2*2) + (2*5) = 14
Total object points=33+14=47
Percentage reuse = 0
NOP = (object point) * [(100- %reuse)/100]
= 47*[(100-0)/100]
= 47
Assuming Developers' experience and environment maturity to be nominal,
Productive rate i.e. PROD=13
NOW,
Estimated Effort=NOP/PROD
=47/13
=3.615 person-month

ER DIAGRAM


Payment_ID
PAYMENT
Payment_status
Passenger_ID
Date_of_payment
Card_number
PAY
Passenger_ID
Passenger_Name
Phone Number
Seat Number
Password
Flight Number
Email Id
PASSENGER
Flight_attendant_id
Flight attendant name
Password
Phone Number
Gender
FLIGHT ATTENDANT
Phone Number
Password
Admin_ID
Admion_Name
ADMIN
REGISTER
TAKE
GIVE
Location_name
Longitude
LOCATION
ENTER
Latitude
Report_date
GET
REPORT
Report_time
Report_date
Report_status
GIVE
REVIEWS
Passenger_Name
Passenger_Email ID
Comment
Rating
ORDERS
Seat_no
Order Number
Bill
Order_time
no_of_dish
Dish_code
Payment Status
UPLOAD
MENU LIST
Price
Dish_code
Quantity
GAMES
Game_id
Game_name
Description
MUSIC
Music_id
Title
Language
Genre
VIDEOS
Movie_id
Category
Language
Description
Genre
Title

DATA DICTIONARY

LOCAL DATABASE

ADMIN

S.no	Field Name	Data Type	Null	Description
1.	Admin_ Name	Char(50)	No	Admin_Name=store the name of admin
2.	Admin_ID	varchar(10)	No	Primary key [Uniquely identify the admin]
3.	Password	varchar(10)	No	Used for Authenticating an admin.
4.	Phone Number	Integer	No	Store Phone number of admin

PASSENGER

S.no	Field Name	Data Type	Null	Description
1.	Passenger_Name	Char(50)	No	Passenger_Name=store the name of passenger
2.	Passenger_ID	varchar(10)	No	Primary key [Uniquely identify the Passenger]
3.	Password	varchar(10)	No	Used for Authenticating a passenger.
4.	Phone Number	Integer	No	Store Phone number of passenger.
5.	Flight Number	Short int	No	It tells about the flight number of the passenger.
6.	Seat Number	Short int	No	It tells about the seat number of the passenger.
7.	Email ID	varchar(50)	No	Passenger’s email[Every passenger have a unique email]

Flight Attendant

S.no.	Field Name	Data Type Data Size	Null	Description
1.	Flight attendant name	Varchar(25)	No	Name of the Flight attendants.
2.	Flight attendant ID	Varchar(25)	No	Primary key [Uniquely identify the flight attendant]
3.	Password	varchar (50)	No	For authenticating of the passenger.
4.	Phone Number	Long int(10)	No	Store Phone number of flight attendant.
5.	Gender	Char(2)	No	Tells about the gender of the passenger.

Menu List

S.no	Field Name	Data Type	Null	Description
1.	Dish_code	Varchar(25)	No	Code given to a particular dish. Primary key[uniquely identifies the dish]
2.	Price	Long int	No	Cost of each dish.
3.	Quantity	Long int	No	Tells us the quantity of food item.

Reviews

S.No	Field Name	Data Type	Null	Description
1.	Passenger_Name	Varchar(50)	No	Passenger_Name=store the name of passenger
2.	Passenger_Email ID	Long int	No	Primary key[uniquely identifies the order]
3.	Rating	Int	No	Give Points to rate the facilities.
4.	Comment	Varchar(100)	Yes	Describe the passenger view about the facilities provided by the software.

Orders

S.no	Field Name	Data type	Null	Description
1.	Order Number	varchar(10)	No	Primary key[uniquely identifies the order]
2.	Order time	time	No	Time at which order placed
3.	Dish_code	varchar(10)	No	Code given to a particular dish
4.	Bill	Integer	No	Your final total will be treated as bill.
5.	No-of-dish	Long int	No	Tells us the quantity of food item.
6.	Seat no	Short int	No	Tells about the seat number of the passenger.
7.	Payment Status	char(20)	No	Tells us that payment is paid or not.

Videos

S.No	Field name	Data Type	Null	Description
1.	Title	Char(50)	No	Name of the video or movie.
2.	Category	varchar(10)	No	It stores the category of the entertainment hub. It can be movie or a video.
3.	Language	varchar(10)	No	Tells about the language of the particular video.
4.	Genre	Varchar(25)	No	It tells about the genre of video like funny, action, romantic etc.
5.	Movie id	int	No	Unique id for each video(payment)
6.	Description	Varchar(100)	Yes	Summarize the video

Music

S.No	Field name	Data Type	Null	Description
1.	Title	Char(50)	No	Name of the video or movie.
2.	Language	varchar(10)	No	Tells about the language of the particular video.
3.	Genre	Varchar(25)	No	genre of music like pop, melody etc.
4.	Music_id	int	No	Unique id for each video

Games

S.No	Field name	Data Type	Null	Description
1.	Game_id	int	No	Unique id for game
2.	Game_name	varchar(10)	No	Name of the game.
3.	Description	Varchar(100)	Yes	Summarize the game

WEATHER DATABASE

Location

S.no	Field Name	Data type	Null	Description
1.	Location name	varchar(10)	No	Tells about the place of which weather has to be reported
2.	Latitude	Int	No	Tell the exact coordinates of location
3.	Longitude	Int	No	Tell the exact coordinates of location
4.	Report date	date	No	The date on which that report is recorded (primary key and foreign key)

Report

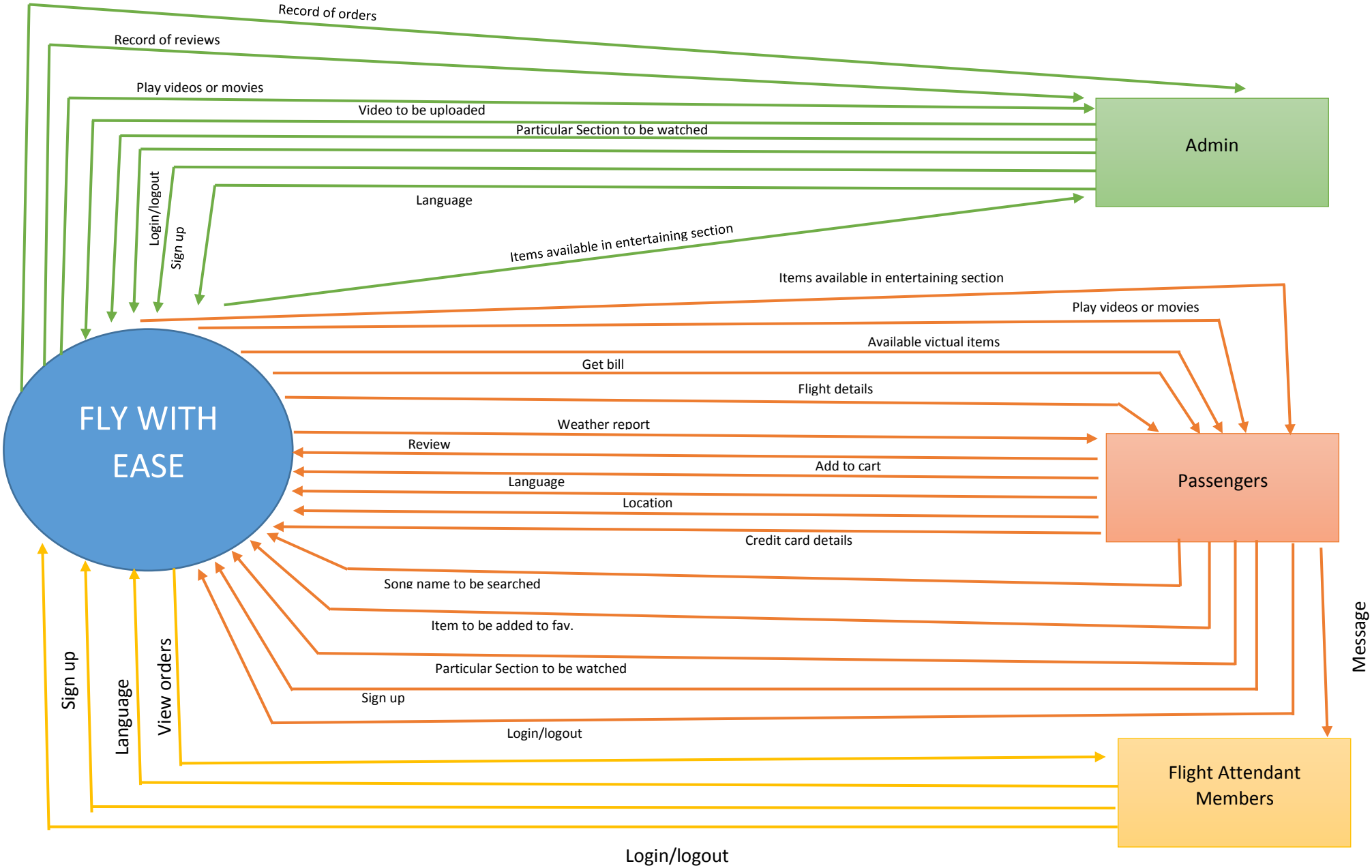
S.no	Field Name	Data type	Null	Description
1.	Report Status	varchar(10)	No	It describes about the current status of the weather at that particular location
2.	Report time	Time	No	The time at which that report is recorded (primary key)
3.	Report date	date	No	The date on which that report is recorded (primary key)

PAYMENT

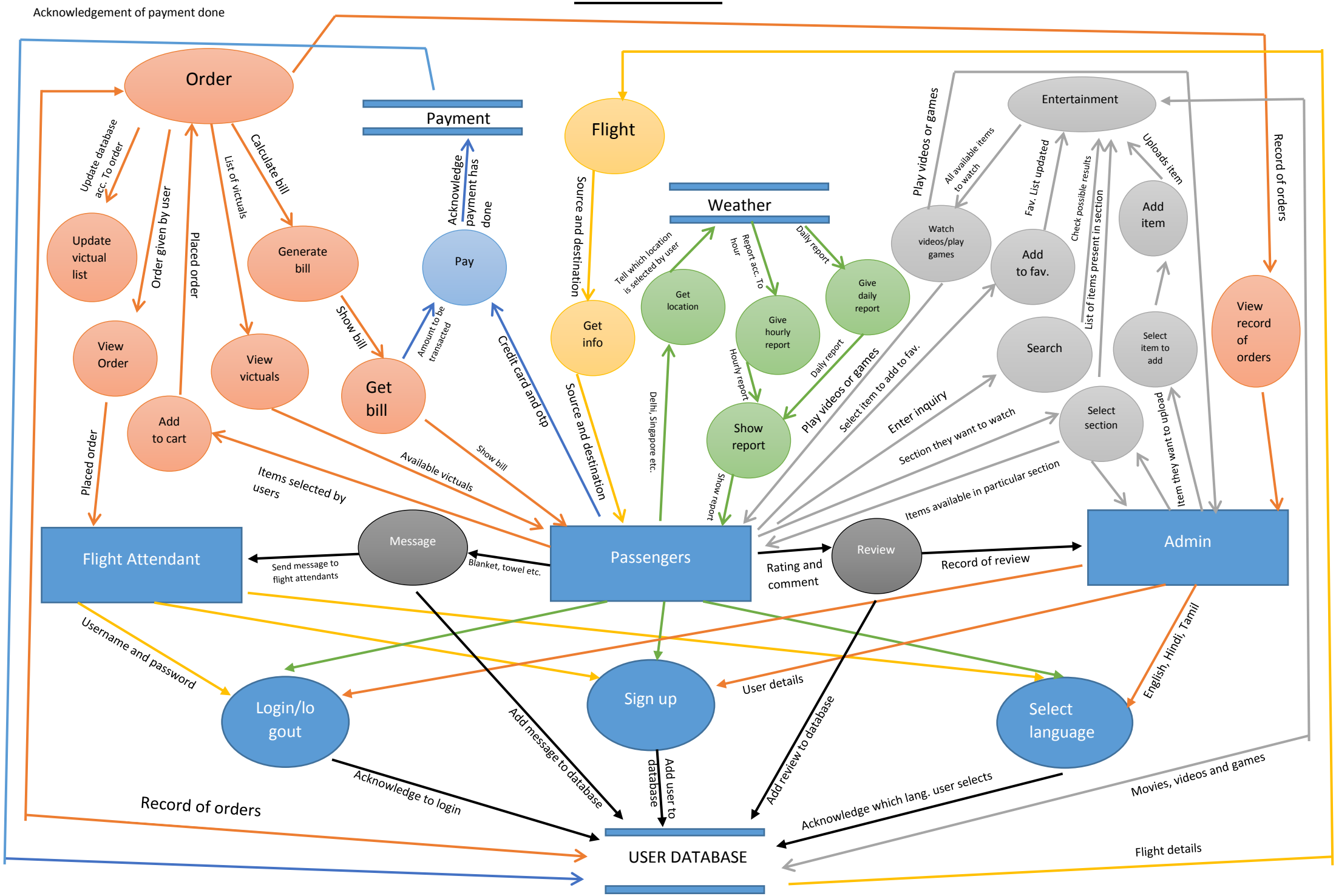
Payment

S.No	Field name	Data Type	Null	Description
1.	Payment id	Int	No	Unique id for each payment(primary key)
2.	Passenger id	Int	No	Tells which passenger paid the payment
3.	Payment status	Bool	No	Tell whether the payment is done or not
4.	Date of payment	Date	No	The date on which payment has done
5.	Card no.	Long int(16)	No	From which card passenger paid the payment

DFD LEVEL-0

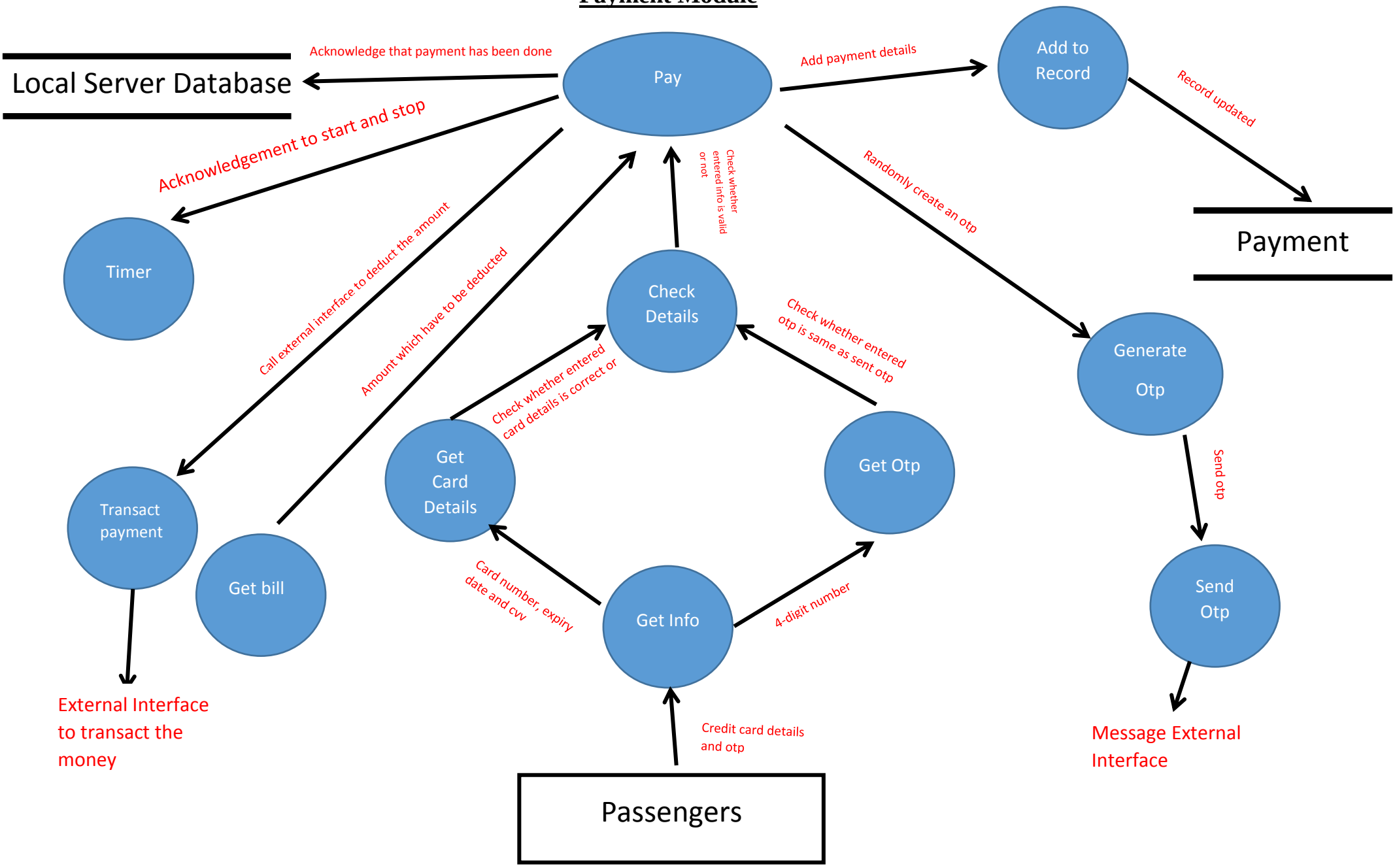

Record of orders
Record of reviews
Play videos or movies
Video to be uploaded
Particular Section to be watched
Login/logout
Sign up
Language
Items available in entertaining section
Admin
Items available in entertaining section
Play videos or movies
Available victual items
Get bill
Flight details
Weather report
Review
Add to cart
Language
Location
Credit card details
FLY WITH EASE
Passengers
Song name to be searched
Item to be added to fav.
Particular Section to be watched
Sign up
Login/logout
Message
Sign up
Language
View orders
Flight Attendant Members
Login/logout

DFD LEVEL-1


Acknowledgement of payment done
Order
Payment
Flight
Weather
Entertainment
Update database acc. To order
Order given by user
List of victuals
Calculate bill
Acknowledge payment has done
Source and destination
Tell which location is selected by user
Report acc. To hour
Daily report
Play videos or games
All available items to watch
Fav. List updated
Check possible results
Uploads item
Record of orders
Update victual list
Generate bill
Pay
Get info
Get location
Give hourly report
Give daily report
Watch videos/play games
Add to fav.
Add item
View Order
View victuals
Get bill
Show bill
Amount to be transacted
Credit card and otp
Source and destination
Delhi, Singapore etc.
Hourly report
Daily report
Play videos or games
Select item to add to fav.
Search
List of items present in section
Select item to add
View record of orders
Placed order
Add to cart
Available victuals
Show bill
Items selected by users
Show report
Enter inquiry
Section they want to watch
Items available in particular section
Select section
Item they want to upload
Flight Attendant
Message
Passengers
Review
Admin
Send message to flight attendants
Blanket, towel etc.
Rating and comment
Record of review
Username and password
Login/logout
Sign up
Select language
User details
English, Hindi, Tamil
Add message to database
Add user to database
Add review to database
Acknowledge to login
Acknowledge which lang. user selects
Movies, videos and games
Record of orders
USER DATABASE
Flight details

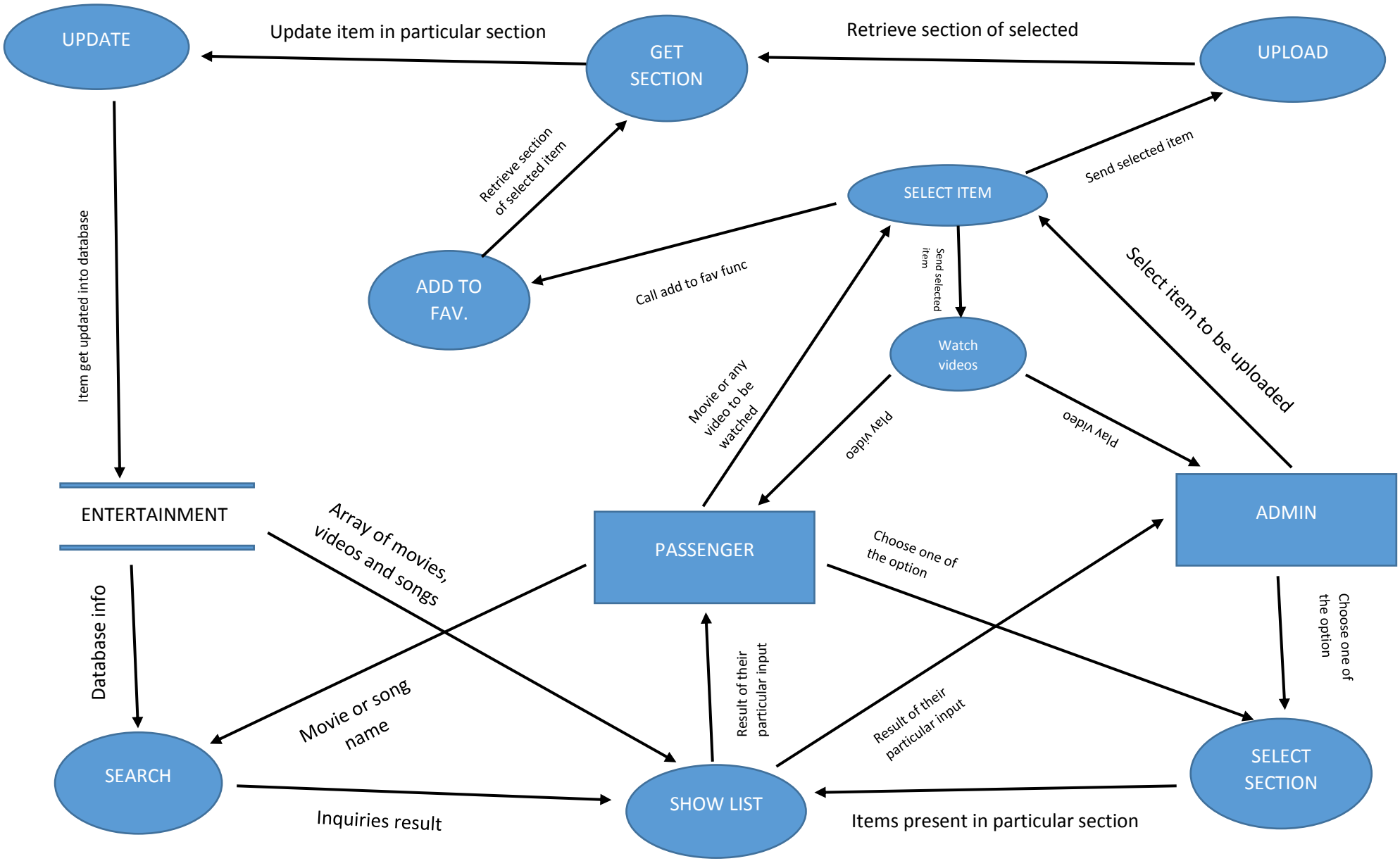
DFD LEVEL-2

Payment Module

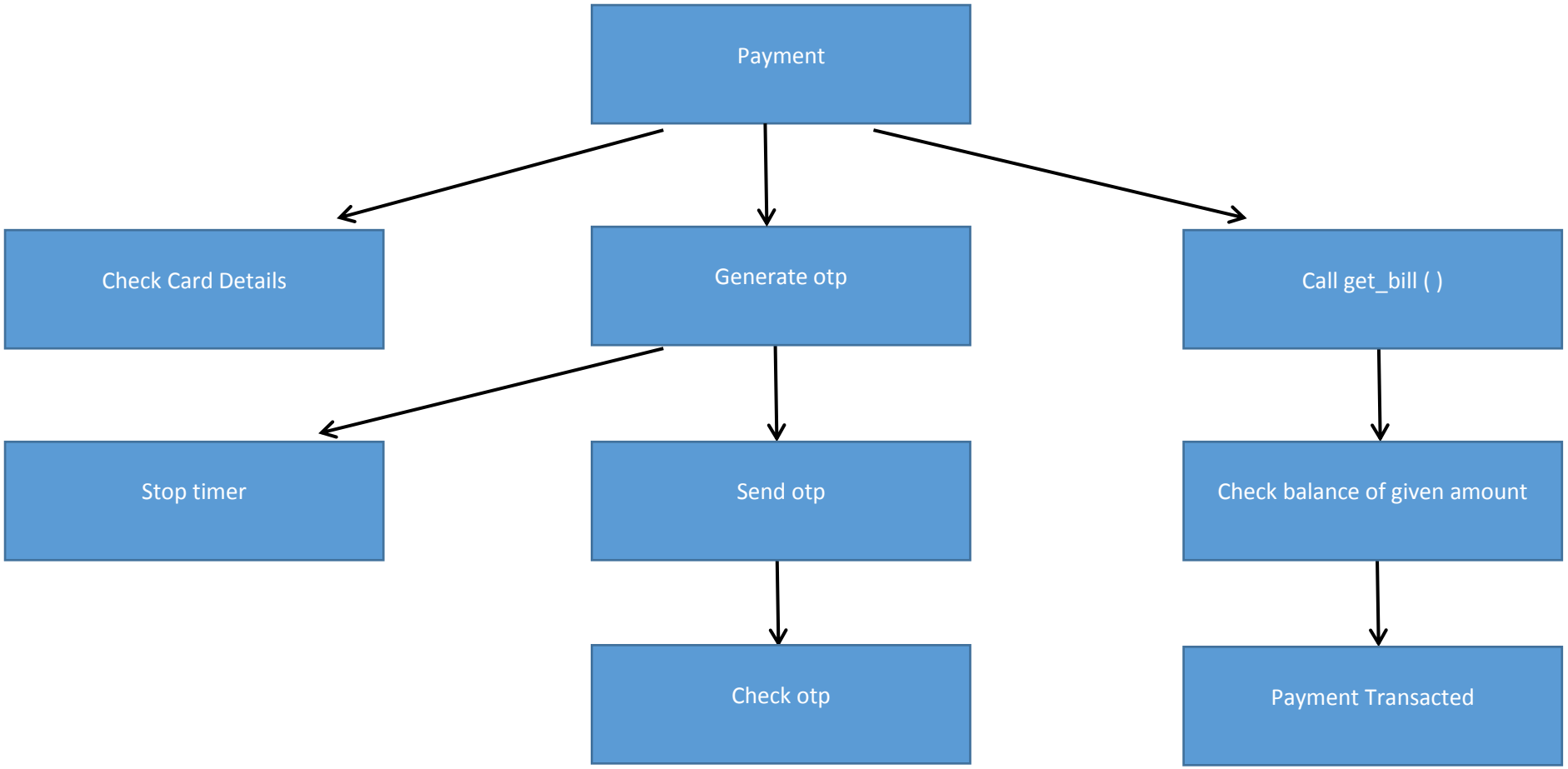


DFD LEVEL-2

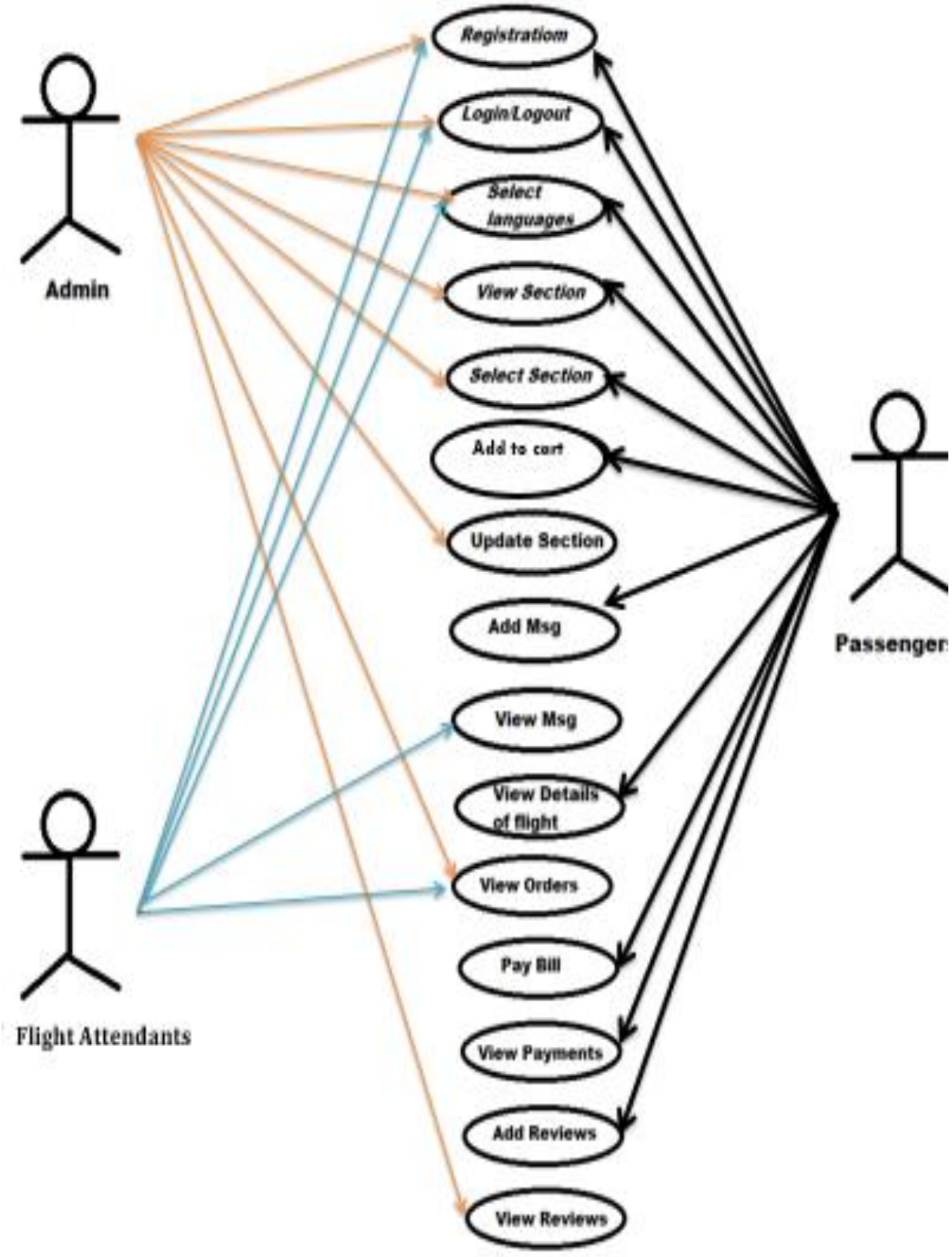
Entertainment Module



ARCHITECTURAL DESIGN



USE CASE DIAGRAM



USE CASE DESCRIPTION

1. Registration

1.1 Introduction
This use case document used to register all the actors of the software.

1.2 Actors
Admin
Flyers
Flight attendants

1.3 Pre-condition
None

1.4 Post-condition
If the registration done successfully then the actors can login themselves.

1.5 Flow of events

Basic flow
All actors will add their details respectively and register themselves.

Admin: admin name, password, contact number and admin-id.
Flight Attendant: username, password, contact number, crew id and gender.
Passengers: username, password, flyer id, flight id, seat number, contact number and email-id.

Alternative flow
If registration is unsuccessful then system displays an error message that could be actor is already registered or please enter a strong password. After the error message, actor can re-register themselves with a new username or strong password.

1.6 Special Requirements
None

1.7 Related use cases
None

2. Login/Logout

2.1 Introduction
This use case document use for login/logout by admin, flyers and flight attendants.

2.2 Actors
Admin
Flyers
Flight Attendants

2.3 Pre-condition
•Actors are already registered.

2.4 Post-condition
After login successfully all actors can access the system.

2.5 Flow of events

Basic flow
• Actors have to enter their login id
• Enter password.
• Then press the button login.
• Now they can access the system.

Alternative flow
If the actors entered the incorrect password or login id then the system shows an error message login failed. Now actors can go to the beginning of basic flow or they can exit from the system.

2.6 Special Requirements
None

2.7 Related use cases
None

3. Select Language

3.1 Introduction
This use case document describe that users can select their desired language.

3.2 Actors
Admin
Flyers
Flight Attendants

3.3 Pre-condition
Actors logged in into the system.

3.4 Post-condition
Actors can access the system in their desired language.

3.5 Flow of events

Basic flow
They can view the options of various languages and can select their desired one.

Alternative flow
If they do not select the language then by default English language will be selected.

3.6 Special Requirements
None

3.7 Related use cases
None

4. Select Section

4.1 Introduction
This use case describe about selecting a section.

4.2 Actors
Admin
Flyers

4.3 Pre-condition
Actors must be logged into the system.

4.4 Post-condition
None

4.5 Flow of events

Basic flow for Admin
Admin can select the section and go to the option to update something.

Basic flow for flyers
They select the section and go to that respective section.

Alternative flow
Actor can log out from the system and exit.

4.6 Special Requirements
None

4.7 Related use cases
None

5. View sections

5.1 Introduction
This use case document use to view all the available sections like (entertainment hub, victuals section, weather section etc.)

5.2 Actors
Admin
Flyers

5.3 Pre-condition
A section must be selected.

5.4 Post-condition
None

5.5 Flow of events

Basic flow
Actor can view the content of selected section.

Alternative flow
Actor can select another section.

5.6 Special Requirements
None

5.7 Related use cases
None

6. Update Section

6.1 Introduction
This use case document describes about updations of sections.

6.2 Actors
Admin

6.3 Pre-condition
Actor must be logged in.

6.4 Post-condition
None

6.5 Flow of events

Basic flow
• select the item to update (music, movie, video or game).
• select the option to add or to delete.
• Admin can add an item from their storage into the system or delete any item from the system.

Alternative flow
Actor can log out.

6.6 Special requirements
None

6.7 Related use case
None

7. Add message

7.1 Introduction
This use case document describe about send messages to the flight attendants.

7.2 Actors
Flyers

7.3 Pre-condition
Actor must be logged into the system and select the messaging section.

7.4 Post-condition
None

7.5 Flow of events

Basic flow
•select messaging section.
•entered into the messaging section.
•type the message what they need like towel, blanket etc.
•Send the message to the flight attendants.

Alternate flow
Actor can logged out.

7.6 Special Requirements
None

7.7 Related use cases
None

8. View message

8.1 Introduction
This use case document describes about view the messages by flight attendants.

8.2 Actors
Flight Attendants

8.3 Pre-condition
Actor must be logged into the system.

8.4 Post-condition
None

8.5 Flow of events

Basic flow
Now, actors can view the messages and serve them to their respective seat number.

Alternate flow
Actor can log out.

8.6 Special requirements
None

8.7 Related use cases
None

9. View Details of flights

9.1 Introduction
This use case document describe about how actor can view the details of their flight.

9.2 Actors
Flyers

9.3 Pre-condition
Flyers must be logged in to the system.

9.4 Post-condition
None

9.5 Flow of events

Basic flow
•select the detail option.
•view the details.

Alternate flow
None

9.6 Special Requirements
None

9.7 Related use case
None

10. View orders

10.1 Introduction
This use case document describe how flight attendants view the orders given by flyers.

10.2 Actors
Flight Attendants

10.3 Pre-condition
Actors must be logged into the system.

10.4 Post-condition
None

10.5 Flow of events

Basic flow
•view the details of the orders.

Alternate flow
None

10.6 Special Requirements
None

10.7 Related use cases
None

11 Pay Bill

11.1 Introduction
This use case document describe about how the payment can be done by flyers for their orders.

11.2 Actors
Flyers

11.3 Pre-condition
Flyers must logged in.
Flyers must order something.

11.4 Post-condition
None

11.5 Flow of events

Basic flow
•Enter the required details like card holder name, card number, cvv number and then respective otp number and finally pay amount.

Alternate flow
If card details entered by the user is not valid, an error message will be generated that please enter the valid card details.

11.6 Special requirements
None

11.7 Related use cases.
None

12 View payments

12.1 Introduction
This use case document describe that admin can view the record of all payments.

12.2 Actors
Admin

12.3 Pre-condition
Actors must be logged in.

12.4 Post-condition
None

12.5 Flow of events

Basic flow
After payments has done actors can view the record of payments.

Alternate flow
Actor can logged out.

12.6 Special Requirements
None

12.7 Related use case
None

13. Add reviews

13.1 Introduction
This use case document is used by the flyers to add their reviews about the travelling experience in the flight.

13.2 Actors
Flyers

13.3 Pre-condition
User must be logged in.

13.4 Post-condition
User can review only one time.

13.5 Flow of events

Basic flow
•Go to the menu bar.
•Select the option of review.
•Enter your name and email Id.
•Rate the service.
•And you can comment your view as well.

Alternate flow
Actor can log out.

13.6 Special Requirements
None

13.7 Related use cases
None

14. View Reviews

14.1 Introduction
This use case document is used by the admin to view the record of reviews given by the flyers.

14.2 Actors
Admin

14.3 Pre-condition
Admin must be logged in.

14.4 Post-condition
None

14.5 Flow of events

Basic flow
• go to the menu bar.
• select the option of view reviews.
• view the reviews.

Alternate flow
Actor can log out the account.

14.6 Special Requirements
None

14.7 Related use cases
None

FUNCTION – BASED METRICS

The function point (FP) metric can be effectively used for measuring the functionality delivered by a system. Using historical data, the FP metric can then be used to:

1. Estimate the cost or effort required to design, code, and test the software.
2. Predict the number of errors that will be encountered during testing.
3. Forecast the number of components and/or the number of projected source lines in the implemented system.

Function points are derived using an empirical relationship based on direct measures of software's information domain and qualitative assessments of software complexity. Information domain values are defined in the following manner:

1. Number of external inputs (EIs): Each external input originates from a user or is transmitted from another application and provides distinct application-oriented data or control information. Inputs are often used to update internal logical files (ILFs).
2. Number of external outputs (EOs): Each external output is derived data within the application that provides information to the user. In this context external output refers to reports, screens, error messages, etc.
3. Number of external inquiries (EQs): An external inquiry is defined as an online input that results in the generation of some immediate software response in the form of an online output (often retrieved from an Internal Logical Files).
4. Number of internal logical files (ILFs): Each internal logical file is a logical grouping of data that resides within the application's boundary and is maintained via external inputs.
5. Number of external interface files (EIFs): Each external interface file is a logical grouping of data that resides external to the application but provides information that may be of use to the application.

Once these data have been collected, a complexity value is associated with each count. Organizations that use function point methods develop criteria for determining whether a particular entry is simple, average, or complex.

Computing function points

(Assuming all the files, inputs, outputs and queries are simple and average)

Information Domain Value	Count		Simple	Average	Complex	
External Inputs (EIs)	40	X	27*3	13* 04	06	= 133
External Outputs (EOs)	12	X	09*4	03*05	07	= 51
External Inquiries (EQs)	06	X	02*3	00*04	06	= 06
Internal Logical Files (ILFs)	12	X	08*7	04*10	15	= 96
External Interface Files (EIFs)	05	X	03*5	02*07	10	= 29

Count Total =315

The Fi (i =1 to 14) are value adjustment factors based on responses to the following questions:
1. Does the system require reliable backup and recovery?

2. Are specialized data communications required to transfer information to or from the application?

4

3. Are there distributed processing functions?

4

4. Is performance critical?

3

5. Will the system run in an existing, heavily utilized operational environment?

5

6. Does the system require online data entry?

5

7. Does the online data entry require the input transaction to be built over multiple screens or operations?

2

8. Are the ILFs updated online?

5

9. Are the inputs, outputs, files, or inquiries complex?

2

10. Is the internal processing complex?

3

11. Is the code designed to be reusable?

4

12. Are conversion and installation included in the design?

5

13. Is the system designed for multiple installations in different organizations?

5

14. Is the application designed to facilitate change and ease of use by the user?

0

Now we have, Sum (fi) =52

To compute function points (FP), the following relationship is used:

FP = Count Total * [0.65 + 0.01 * Σ (Fi)]

Where count total is the sum of all FP entries obtained from the table given above.

FP = 315*[0.65+ (0.01*52)]

= 315*[0.65+0.52]

= 315*1.17

FP = 368.55 person-month

Risk Analysis

Risk analysis and management are actions that help a software team to understand and manage risk. A risk involves future happening, uncertainty, choice and change.

RISK COMPONENTS

- Performance Risk: - It is the degree of uncertainty that the product will meet its requirements and be fit for its intended use.
- Cost Risk: - It is the degree of uncertainty that the project budget will be maintained.
- Support Risk: - It is the degree of uncertainty that the resultant software will be easy to correct, adapt, and enhance.
  Schedule Risk: - It is the degree of uncertainty that the project schedule will be maintained and that the product will be delivered on time.

RISK IDENTIFICATION

Risk identification is a systematic attempt to specify threats to the project plan.
There are two distinct types of risks for each of the categories general risks and product-specific risks. General risks are a potential threat to every software project and Project-risks can be identified only by those with a clear understanding of the technology, the people, and the environment that is specific to the software that is to be built.
One method for identifying risk is to create a risk item checklist. Checklist can be used for risk identification and focuses on some subset of known and predictable risks in the following generic subcategories:

- Product Size: It is the risk associated with the overall size of the software to be built or modified.
- Business Impact: It is the risks associated with constraints imposed by management or the marketplace.
- Stakeholder Characteristics: It is the risks associated with the sophistication of the stakeholders and the developer's ability to communicate with stakeholders in a timely manner.
- Process Definition: It is the risks associated with the degree to which the software process has been defined and is followed by the development organization.
- Development Environment: It is the risks associated with the availability and quality of the tools to be used to build the product.
- Technology to Be Built: It is the risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
- Staff Size and Experience: It is the risks associated with the overall technical and project experience of the software engineers who will do the work.

Risk Monitoring

The risk management steps can be organised into a separate risk mitigation, monitoring, and management plan (RMMM). Risk Monitoring is a project tracking activity with three primary objectives:

- To assess whether predicted risks do, in fact, occur;
- To ensure that risk aversion steps defined for the risk are being properly applied;
- To collect information that can be used for future risk analysis.

ASSESSING OVERALL PROJECT RISK

The following questions have derived from risk data obtained by surveying experienced software project managers in different parts of the world.

1. Have top software and customer managers formally committed to support the project?
   Yes
2. Are end users enthusiastically committed to the project and the system/product to be built?
   Yes
3. Are requirements fully understood by the software engineering team and its customers?
   Yes
4. Have customers been involved fully in the definition of requirements?
   Yes
5. Do end users have realistic expectations?
   Yes
6. Is the project scope stable?
   Yes
7. Does the software engineering team have the right mix of skills?
   Yes
8. Are project requirements stable?
   Yes
9. Does the project team have experience with the technology to be implemented?
   No
10. Is the number of people on the project team adequate to do the job?
    No
11. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?
    Yes

RISK PROJECTION

Risk projection also called risk estimation, attempts to rate each risk in two ways:

- the likelihood or probability that the risk is real and
- the consequences of the problem associated with the risk, should it occur.

A risk table is a simple technique for risk projection.
For our project, risk table is as follows: -

RISK	**CATEGORY**	**PROBABILITY**	**IMPACT**
Less no. of customers	BR	25%	2
Overloading of orders	TR	30%	1
Security Issues	PR	15%	1
Fail to set up a network	TR	10%	2
High response time	PR	15%	3
Staff Inexperienced	BR	20%	2
Delivery deadlines will be tightened	BR	10%	3

CATEGORY	**IMPACT VALUES**
Catastrophic	1
Critical	2
Marginal	3
Negligible	4

TESTING

WHITE BOX TESTING

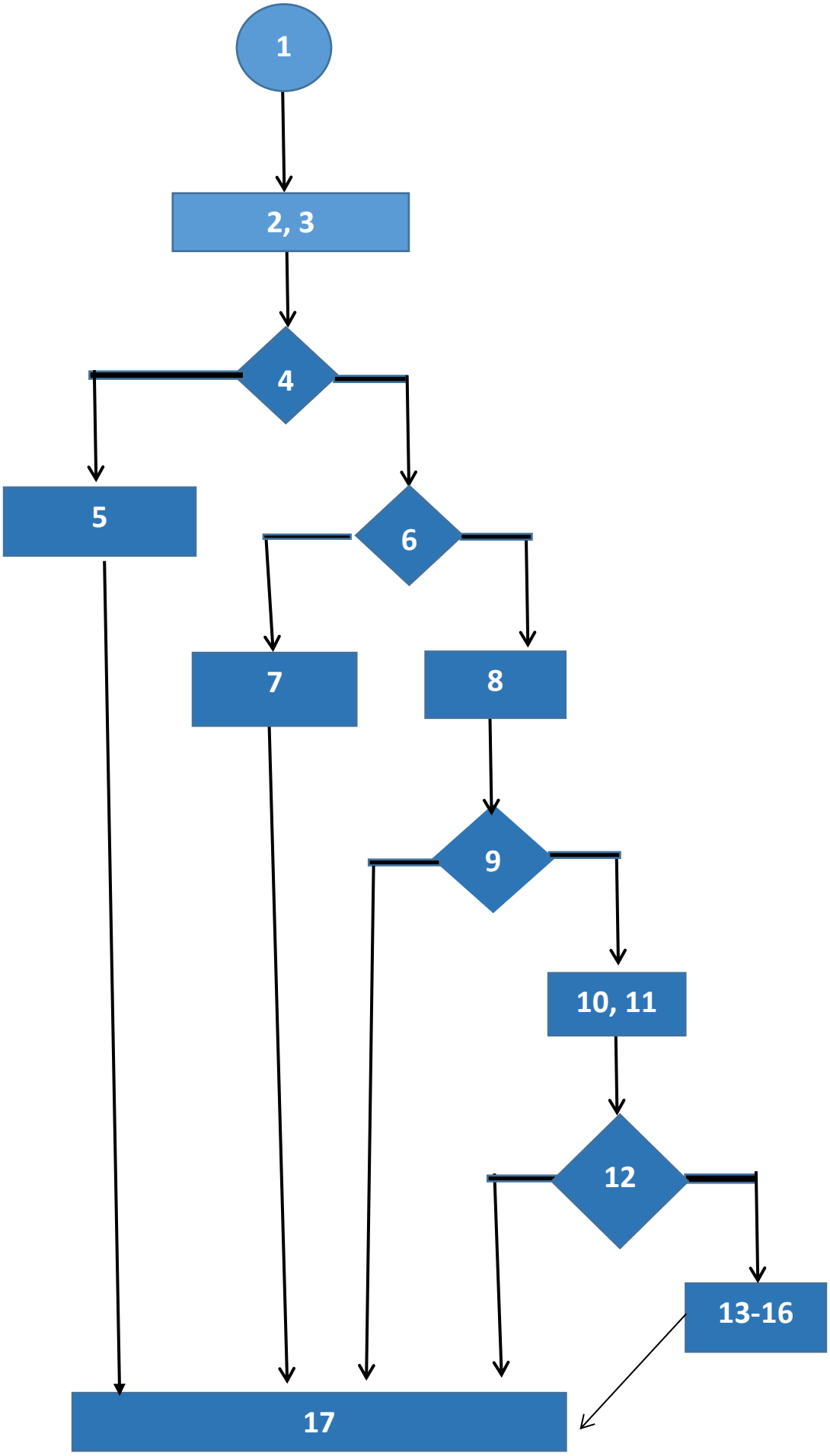
White box testing, also known as Glass-box testing is a test case design philosophy that uses the control structure described as part component-level design to derive test cases. Using white-box testing methods, you can derive test cases that-

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Exercise all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to ensure their validity.

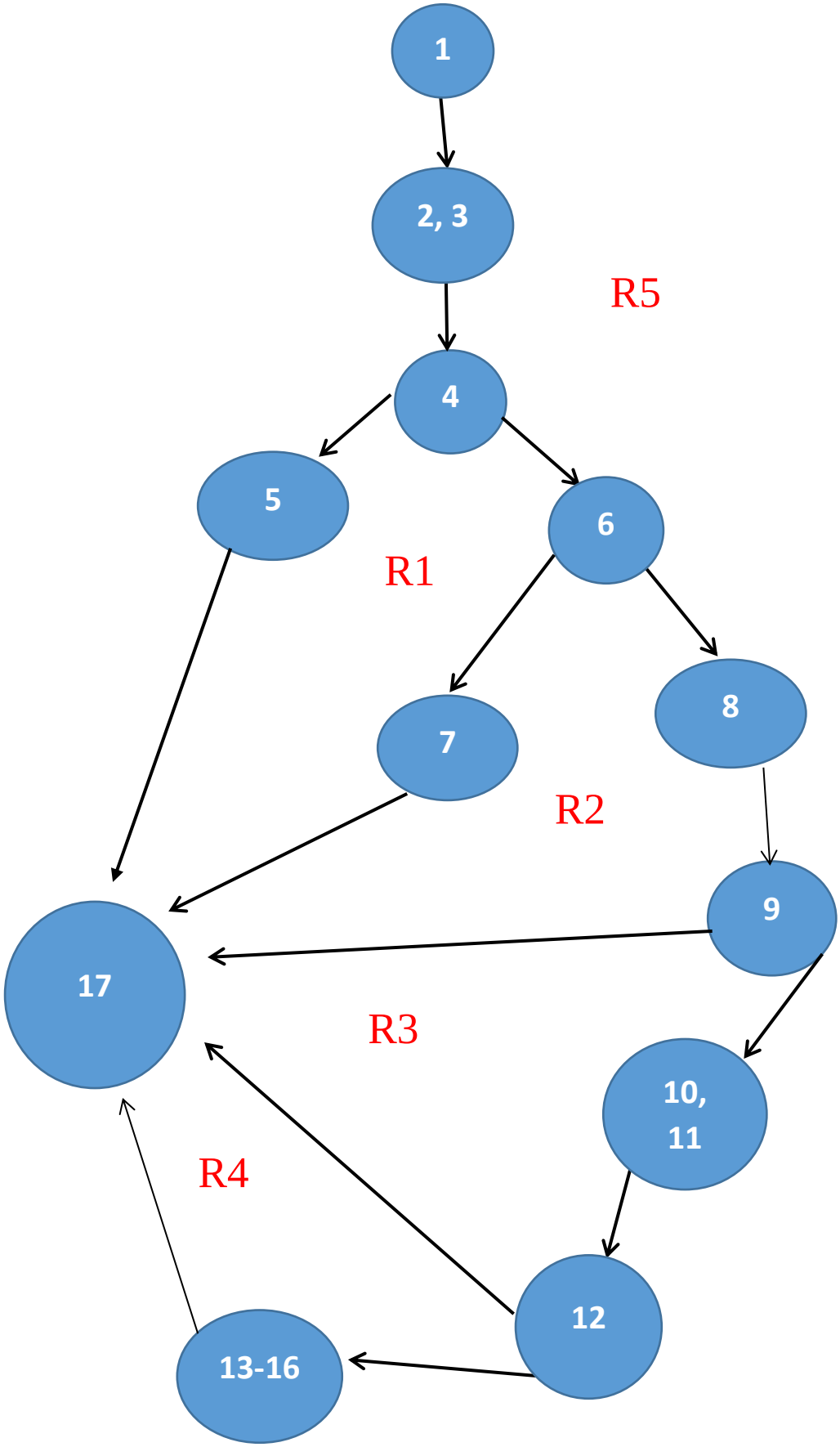
We are performing this white-box testing for payment module:

```
1) int payment_module( )//return 1 if payment done successfully otherwise 0
2) {
3)    Info info.get_info(); /*Info is a class which stores all the information entered by flyers for their payment*/
4)    if(info.card_no.length()!=16)
5)      return 0;
6)    if(info.exp_date<present_date())
7)      return 0;
8)    long otp=generate_otp();
9)    timer_start (900); /*this func. will forcefully close this func if within 900 seconds user doesn't enter the otp*/
10)   long otp1=info.get_otp();
11)   stop_timer();
12)   if(otp1= =otp)
13)   {
14)      int val=payment(get_bill()); /*this func returns 1 if amount successfully debited from given account*/
15)      return val;
16)    }
17) }
```

FLOW CHART



FLOW GRAPH



CYCLOMATIC COMPLEXITY OF RESULTING GRAPH

1. V (G) = No. of regions of the flow graph = 5
2. V (G) = E-N+2
   = 15-12+2
   = 5

Where E= No. of edges of the flow graph and
N= No. of nodes of the flow graph

3. V (G) = P+1
   = 4+1
   =5

Where P= Predicate nodes
Hence by all means cyclomatic complexity is 5.

SET OF LINEARLY INDEPENDENT PATHS

Path 1:1-2-3-4-5-17
Path 2:1-2-3-4-6-7-17
Path 3: 1-2-3-4-6-8-9-17
Path 4: 1-2-3-4-6-8-9-10-11-12-17
Path 5: 1-2-3-4-6-8-9-10-11-12-13-14-15-16-17

TEST CASES

1. Case to check whether card no. entered by user is of length 16 or not.
2. Case to check whether card is expired or not.
3. Case to check whether user entered otp in appropriate time or not.
4. Case to check whether otp entered by user is correct or not.
5. Case to check whether money successfully transacted from account or not.

BIBLIOGRAPHY

- "Software Engineering- A practitioner's approach (7th edition)" by Roger S. Pressman.
- "Software Engineering (Revised 2nd edition)", by K.K. Aggarwal & Yogesh Singh.
- 830-1998-IEEE recommended Practice for Software Requirement Specifications.